

Table des matières

Préface	15
Introduction	17
Jean-Pierre BRIOT et Yves DEMAZEAU	
Chapitre 1. Systèmes multi-agents : principes généraux et applications. . . .	27
Brahim CHAIB-DRAA, Imed JARRAS, Bernard MOULIN	
1.1. Des agents autonomes aux systèmes multi-agents	27
1.1.1. Qu'est ce qu'un agent ?	27
1.1.1.1. Historique	29
1.1.1.2. Les applications des agents autonomes.	32
1.1.2. Les systèmes multi-agents	33
1.1.2.1. Présentation	33
1.1.2.2. Historique	35
1.2. Interactions et coopération entre agents.	37
1.2.1. Interactions entre agents coopératifs	39
1.2.2. Interactions entre agents égocentrés.	40
1.3. Coordination entre agents	41
1.4. Négociation entre agents.	44
1.5. Planification dans un environnement multi-agent.	45
1.6. Communication entre agents	47
1.6.1. Transfert de plans ou de messages.	48
1.6.2. Echange d'informations grâce à un tableau noir	49
1.6.3. Actes de discours et conversations	49
1.6.4. Communication utilisant KQML, ACL et les conversations	52
1.7. Méthodes de conception des systèmes multi-agents	52

10 Principes et architecture des SMA

1.8. Quelques exemples d'application des SMA	55
1.8.1. Application des SMA aux télécommunications	56
1.8.2. Le système ADEPT	56
1.8.3. Le système GUARDIAN	56
1.8.4. Les systèmes d'information coopératifs (SIC)	57
1.9. Conclusion	60
1.10. Bibliographie	60

Chapitre 2. Modèles et architectures d'agents 71

Olivier BOISSIER

2.1. Introduction	71
2.2. Modèles d'agent	72
2.2.1. Architecture/comportement d'agent	73
2.2.2. Descriptions d'un agent	74
2.3. Dimensions des architectures d'agents	75
2.3.1. Vision coordination	76
2.3.1.1. Architecture d'agent autonome	77
2.3.1.2. Architecture d'agent interagissant	77
2.3.1.3. Architecture d'agent social	77
2.3.2. Vision raisonnement	78
2.3.2.1. Architecture d'agent réactif	78
2.3.2.2. Architecture d'agent délibératif	78
2.3.2.3. Architecture d'agent hybride	78
2.4. Architectures d'agents autonomes	79
2.4.1. Architectures d'agents réactifs	79
2.4.1.1. MANTA	79
2.4.1.2. Architecture Subsumption	80
2.4.2. Architectures d'agents délibératifs	81
2.4.3. Architecture d'agent hybride	83
2.5. Architectures d'agents interagissants	84
2.5.1. Architecture d'agent réactif	85
2.5.2. Architectures d'agents délibératifs	86
2.5.2.1. <i>Agent Oriented Programming</i>	87
2.5.2.2. COSY	87
2.5.3. Architectures d'agents hybrides	88
2.6. Architectures d'agents sociaux	90
2.6.1. Architectures d'agents réactifs	90
2.6.1.1. Eco-Résolution	91
2.6.1.2. PACORG	91
2.6.2. Architectures d'agents délibératifs	91
2.6.2.1. ADEPT	92
2.6.2.2. <i>Cooperative Information Agent</i>	93
2.6.2.3. Autres architectures	95

2.7. Architectures.	96
2.7.1. Décomposition modulaire.	97
2.7.2. Interactions intermodules, cycle de contrôle.	97
2.8. Discussion	98
2.9. Bibliographie	99
 Chapitre 3. Modèles de communication	109
Jean-Luc KONING et Sylvie PESTY	
3.1. Introduction	109
3.2. Modes de communication entre agents	111
3.2.1. Communication indirecte <i>via</i> l'environnement	111
3.2.2. Communication directe par passage de messages.	112
3.2.3. Communication par passage de messages et agents	112
3.3. De l'échange des connaissances aux langages.	114
3.3.1. La philosophie du langage ordinaire	114
3.3.1.1. La notion d'acte de langage	114
3.3.1.2. La théorie des actes de langage	115
3.3.2. Langages de communication	117
3.3.2.1. Le langage KQML.	117
3.3.2.2. Le langage ACL	119
3.3.2.3. En résumé	123
3.4. Structuration des échanges : les protocoles d'interaction	123
3.4.1. Description de l'approche par protocole	124
3.4.2. Représentation et spécification des protocoles	125
3.4.3. Principaux protocoles d'interaction	128
3.4.3.1. <i>Contract Net Protocol</i>	128
3.4.3.2. Protocoles d'introduction.	129
3.4.3.3. Autres protocoles.	130
3.5. Conclusions et perspectives.	131
3.6. Bibliographie	132
 Chapitre 4. Modèles de coordination d'agents cognitifs	139
Amal EL FALLAH-SEGHRUCHNI	
4.1. Problématique de la coordination	139
4.2. Perspectives de la coordination.	140
4.2.1. Pluralisme	140
4.2.2. Recherche de la cohérence globale du système	141
4.3. Scénario des agents déménageurs	143
4.4. Dimensions de la coordination	144
4.4.1. Structure d'un agent	144
4.4.2. Modèle de spécification du problème de coordination.	145

4.4.3. Mécanismes de coordination :	
modèles orientés tâches <i>versus</i> modèles orientés agents	148
4.4.3.1. Modèles de coordination orientés tâches.	148
4.4.3.2. Modèles de coordination orientés agents	149
4.4.3.3. Mécanisme de coordination	149
4.4.3.4. Exemple de coordination de plans	149
4.5. Approches de coordination d'agents.	151
4.5.1. Coordination basée sur la résolution distribuée de problèmes	151
4.5.2. Coordination basée sur les structures organisationnelles	153
4.5.2.1. Organisations statiques	154
4.5.2.2. Organisations dynamiques	154
4.5.3. Coordination basée sur les protocoles de coopération	155
4.5.4. Modèles de négociation et de prise de décision distribuée	156
4.5.4.1. Travaux basés sur la théorie des jeux.	156
4.5.4.2. Travaux basés sur la théorie d'aide à la décision.	158
4.5.5. Modèles de coordination fondés sur la formation de coalitions	159
4.6. Coordination d'actions par planification multi-agent et distribuée. . . .	162
4.6.1. La coordination par synchronisation d'actions	162
4.6.2. La planification multi-agent comme spécialisation de la résolution distribuée de problème	162
4.6.3. La coordination par planification multi-agent centralisée ou distribuée.	163
4.6.4. Planification multi-agent centralisée et plans distribués.	163
4.6.5. Planification multi-agent distribuée.	164
4.6.6. Processus de coordination d'actions par planification	167
4.6.6.1. Caractérisation des situations d'interaction	167
4.6.7. Exécution concurrente de plans et dynamique de l'environnement .	170
4.7. Critères d'évaluation des systèmes de coordination.	171
4.8. Conclusion	172
4.9. Bibliographie	172
Chapitre 5. Environnements de développement	177
Zahia GUESSOUM et Michel OCCELLO	
5.1. Introduction	177
5.2. Des défis pour les environnements de développement multi-agents . . .	178
5.3. Vers une généricité de l'opérationnalisation des SMA.	180
5.3.1. Des langages de programmation.	180
5.3.1.1. Un langage d'agent : AGENT0	181
5.3.1.2. Panorama des langages multi-agents	182
5.3.1.3. Discussion.	185
5.3.2. Des architectures logicielles	186
5.3.2.1. Panorama des architectures logicielles génériques utilisées en SMA.	186

5.3.2.2. Discussion	188
5.3.3. Des environnements de développement	189
5.3.3.1. Panorama des environnements de développement multi-agents	190
5.4. Exemples d'environnements de développement	192
5.4.1. MACE	192
5.4.1.1. Agents	192
5.4.1.2. Outils de développement	193
5.4.1.3. Applications	193
5.4.2. DIMA (développement et implémentation SMA)	194
5.4.2.1. Outils de développement	195
5.4.2.2. Agents	196
5.4.2.3. Applications	196
5.4.3. MASK (<i>Multi-Agent System Kernel</i>)	197
5.4.3.1. Agents	197
5.4.3.2. Outils de développement	197
5.4.3.3. Applications	199
5.4.4. Zeus	199
5.4.4.1. Agents	199
5.4.4.2. Outils de développement	199
5.4.4.3. Applications	200
5.4.5. Discussion	200
5.5. Vers des AGL multi-agents	201
5.6. Bibliographie	202

Chapitre 6. Application des systèmes multi-agents

dans les télécommunications	207
--	------------

Thierry BOUDON

6.1. Introduction	207
6.1.1. Evolutions des télécommunications	207
6.1.2. Intérêts des SMA pour les télécommunications	208
6.1.3. Historique des recherches sur les SMA dans les télécommunications	209
6.2. Applications des systèmes multi-agents dans le <i>web</i>	209
6.2.1. Services d'assistance dans l'organisation et le déroulement de voyages	209
6.2.2. Extensions de ces expérimentations au commerce électronique	213
6.2.3. Application des agents à la création de communautés <i>web</i>	216
6.3. SMA dans les services et réseaux de télécommunication	218
6.3.1. Application des SMA aux réseaux privés virtuels	218
6.3.2. Utilisation de SMA dans le cadre des mobiles de troisième génération et du RI	221
6.3.3. Application des SMA à la supervision et gestion de réseaux	225
6.4. SMA dans l'ingénierie logicielle des télécommunications	227

14 Principes et architecture des SMA

6.4.1. FIPA : un standard d'interopérabilité agent	227
6.4.2. Méthodologie, ateliers de conception, de développement et de validation	229
6.5. Conclusion	231
6.6. Bibliographie	233

Chapitre 7. Systèmes multi-agents et écosystèmes. 235
François BOUSQUET et Christophe LE PAGE

7.1. Introduction	235
7.2. Ecologie et modélisation de l'individu	236
7.2.1. De l'équilibre à l'organisation	236
7.2.2. Les modèles basés sur l'individu	240
7.3. SMA et modélisation des écosystèmes	243
7.3.1. Agents, environnement, interactions, organisation	245
7.3.1.1. Agents	245
7.3.1.2. Interactions et environnement	249
7.3.1.3. Organisation : le spatial et le social.	250
7.3.2. Plates-formes	252
7.3.3. Objectif : propriétés génériques ou représentation des connaissances ?	253
7.4. Quelques éléments de discussion.	255
7.5. Conclusion	258
7.6. Bibliographie	258

Index 267

Préface

Dans le monde des chercheurs en systèmes multi-agents (SMA) et en intelligence artificielle distribuée (IAD), la communauté francophone a été l'une des premières, une des plus intégratives et des plus stimulantes. L'activité française en SMA et IAD a commencé à être très visible internationalement au milieu des années 1980. En effet, plusieurs thèmes en SMA se sont développés spécifiquement en France. Selon moi, une des forces de cette école française a été la capacité d'intégrer des perspectives provenant de plusieurs disciplines, en utilisant les résultats des unes pour faire avancer de nouveaux thèmes et problèmes dans d'autres. Par exemple, les chercheurs de la communauté francophone ont été parmi les premiers à étudier les rapports étroits entre les idées en matière de programmation à objets, en programmation concurrente et répartie, en représentation des connaissances et en systèmes multi-agents. Le rassemblement des courants de la représentation des connaissances par *frames* et objets, avec les objets concurrents et distribués dotés d'activités autonomes, a donné naissance à une certaine idée des agents. Ainsi, l'idée d'agents a été créée à partir d'une orientation forte, utilitaire, ascendante et pratique en matière de programmation. Les agents sont le résultat logique d'une approche unifiant plusieurs courants de recherche et de développement préexistants et très actifs. Ceci invalide certaines critiques qui déclarent que l'agent est un argument marketing ou simplement un nouveau nom donné à l'IA. Non ! Dans cette perspective, qui est principalement venue de France à la fin des années 1980, un agent est de la connaissance dotée de structures collectives d'action (les objets distribués). Les SMA n'ont pas nécessairement besoin d'être fondés sur la base d'une résolution distribuée de problèmes, ce qui a longtemps été l'opinion dominante aux Etats-Unis.

Un deuxième type d'intégration réalisé par la communauté française a été la tentative d'unifier les agents à grain fin et à gros grain. Les chercheurs français ont d'abord cherché à composer de plus grandes structures d'agents à partir de composants à grain fin, et à concevoir des problèmes distribués à très grande échelle

mais à grain fin (tels que les mécanismes de vision de bas niveau) en tant que systèmes multi-agents. Ces problématiques de compositionnalité des interactions étaient critiques, aussi bien pour envisager des systèmes de connaissances distribués à large échelle et des objets concurrents distribués, que pour la communauté agent. Pour cette raison, un courant de recherche a émergé à partir de plusieurs groupes de recherche travaillant dans ces domaines au départ distincts. Il a donné naissance à des modèles plus puissants et permis de poser des fondations communes aux trois communautés.

Une troisième intégration a tenté d'unifier clairement les concepts d'architectures d'agents réactives et délibératives. Un certain nombre de chercheurs travaillant en France ont commencé au début des années 1990 à construire des systèmes multi-agents délibératifs émergents, ainsi que des structures complexes de résolution de problèmes à partir de composants réactifs. En outre, ils ont montré l'application pratique qui pouvait être faite de ces systèmes multi-échelles ainsi que les manières dont ils pouvaient être employés pour représenter des systèmes biologiques et physiques complexes, aussi bien que des problèmes de calcul. En conclusion, les chercheurs français ont initié certaines des premières tentatives de méthodologies systématiques de construction de logiciels à base d'agents, résultats directs de leurs travaux d'unification.

Personnellement j'ai eu le plaisir de travailler en France pendant plusieurs années, et de collaborer avec plusieurs des chercheurs francophones en SMA qui ont été parmi les pionniers dans ces idées. J'ai appris beaucoup de ces collaborations. Deux des penseurs les plus actifs et les plus influents dans les sujets mentionnés ci-dessus, Jean-Pierre Briot et Yves Demazeau, ont rassemblé un certain nombre de leurs collègues francophones pour synthétiser les connaissances actuelles sur les systèmes multi-agents dans une perspective francophone. Ce livre en est le résultat. C'est un travail profond et réfléchi, couvrant une sélection des sujets au centre des SMA ainsi que certains sujets plus spécialisés. Avec d'autres ouvrages francophones importants couvrant le champ des SMA (je pense à des publications telles que *Les Systèmes Multi-Agents* de Jacques Ferber et *Conscience Artificielle et Systèmes Adaptatifs* d'Alain Cardon), ce travail rend accessible la recherche moderne en matière d'agents. Je suis donc très heureux de voir la publication de cette contribution de qualité à la compréhension des systèmes multi-agents.

Les GASSER
Urbana, Illinois, août 2001

Introduction

L'informatique est en train de changer de manière assez profonde. Tout d'abord, elle devient ubiquitaire. Au départ confinée aux ordinateurs, elle est en train d'investir les objets de la vie courante : téléphones portables, assistants personnels, maison, etc. Elle devient ainsi de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer. La décentralisation est donc la règle et une organisation coopérative entre modules logiciels est un besoin. De plus, la taille, la complexité et l'évolutivité croissantes de ces nouvelles applications informatiques font qu'une vision centralisée, rigide et passive (contrôlée explicitement par le programmeur) atteint ses limites. On est ainsi naturellement conduit à chercher à donner plus d'autonomie et d'initiative aux différents modules logiciels. Le concept de système multi-agent propose un cadre de réponse à ces deux enjeux complémentaires (et à première vue contradictoires) : l'autonomie et l'organisation.

Cet ouvrage a pour ambition de présenter les principes, techniques et exemples d'applications des systèmes multi-agents. Il est complémentaire au livre de Jacques Ferber [FER 95], premier ouvrage en français présentant le domaine et paru en 1995. Notre ouvrage développe les concepts et techniques de base des systèmes multi-agents ainsi que de premiers domaines d'application (par exemple, le commerce électronique). Il ne développera pas la notion d'agent mobile ni celle d'agent assistant d'interface (les raisons en seront précisées ci-dessous). Un deuxième ouvrage à paraître après celui-ci développera des aspects méthodologiques complémentaires et d'autres applications, par exemple en robotique.

Cette introduction a pour objectif de montrer l'importance et de situer rapidement le concept de système multi-agent (en particulier vis-à-vis de la

programmation classique). Elle ne rentrera pas dans le détail des concepts et techniques (concepts d'architectures d'agents, de coordination, d'organisations, etc.) qui seront introduits dans les différents chapitres.

Qu'est-ce qu'un agent ?

Il n'existe pas encore un consensus sur la définition d'un agent. En plus de la relative jeunesse du domaine, cela s'explique par le fait que diverses communautés revendiquent ce terme avec des problématiques assez différentes (par exemple, en ce qui concerne les agents mobiles) même si celles-ci sont complémentaires et conduites à se rencontrer à terme.

Nous proposons une première définition suivante : « Un agent est une entité logicielle ou physique à laquelle est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents ». Quelques définitions alternatives seront données au début du chapitre 1.

Qu'est-ce qu'un système multi-agent ?

La définition d'un système multi-agent (avec son acronyme SMA, et MAS pour *Multi-Agent System* en anglais) est plus immédiate : « Un système multi-agent est un ensemble organisé d'agents ». Nous ne faisons que suivre ici la définition usuelle du terme système : « Un ensemble organisé d'éléments ». Cela signifie que dans un système multi-agent, il existe une ou plusieurs organisations qui structurent les règles de cohabitation et de travail collectif entre agents (définition des différents rôles, partages de ressources, dépendances entre tâches, protocoles de coordination, de résolution de conflits, etc.). Dans un même système, il existe en général plusieurs organisations et un même agent peut appartenir à plusieurs simultanément. Une organisation économique telle qu'une entreprise, mais aussi une organisation animale telle qu'une fourmilière, sont des exemples d'organisations d'agents dans le monde réel. Suivant les cas, les comportements des agents sont plus ou moins complexes et rationnels et l'organisation est plus ou moins adaptative. Central aux systèmes multi-agents est l'équilibre (et la complémentarité) entre autonomie et organisation.

Les agents sont en général situés dans un environnement (par exemple, topologique) contenant également des entités passives, manipulées par les agents (par exemple, des ressources, des données, des objets physiques...) et communément appelées *objets*. Chaque agent n'a qu'une connaissance partielle de son environnement et des autres agents. Un système multi-agent est donc intrinsèquement décentralisé.

Un premier exemple

Comme premier exemple, considérons le projet d'envoyer un robot sur une planète lointaine pour l'explorer. Le délai de communication avec la Terre est tel qu'il est difficile d'envisager un contrôle direct à partir du sol. Il est donc indispensable de doter ce robot d'autonomie et d'initiative, de manière à prendre des décisions en fonction de la situation locale avec une réactivité suffisante. Ceci n'empêche pas, bien sûr, de maintenir une communication régulière avec le sol pour recevoir des informations et si besoin réajuster la mission.

Plutôt que d'envoyer un seul robot multispécialisé (locomotion, vision, prélèvement d'échantillons, analyse, communication avec la Terre, etc.), il peut être intéressant d'y substituer plusieurs robots. Ils pourront être spécialisés (par exemple, un robot spécialisé dans les prélèvements, un autre dans les analyses – ce dernier étant éventuellement immobile –, etc.). Mais ils devront coopérer et coordonner leurs activités à partir de leurs connaissances et comportements locaux. Une telle spécialisation selon les expertises rend une conception et une construction plus aisées qu'un robot unique qui doit savoir tout faire. Une telle organisation offre également une meilleure efficacité potentielle (du fait des tâches pouvant être effectuées en parallèle). De plus, si l'on assure une certaine redondance (par exemple : plusieurs robots d'analyse), cela confère une certaine robustesse à l'ensemble du système en cas de panne d'un de ses éléments. Ceci est donc un premier exemple de système multi-agent. Dans ce cas, les agents sont robotiques (on parle de robotique collective). Un autre exemple est celui de la surveillance de réseaux dans laquelle on délègue des tâches de surveillance, détection d'anomalies, diagnostic et réparations, à différents agents logiciels coopératifs.

Applications

Les domaines d'application privilégiés des systèmes multi-agents sont par exemple le commerce électronique (voir le chapitre 6), le contrôle de processus et la robotique collective. Ainsi, les principaux industriels des télécommunications dans le monde animent une initiative de standardisation des agents et en premier lieu de leur interopérabilité : la Foundation for Intelligent Physical Agents (www.fipa.org). Ceci sera notamment détaillé dans le chapitre 3. D'autres domaines d'application se concrétisent, tels les loisirs, la simulation de phénomènes complexes (voir le chapitre 7), etc.

Positionnement historique

On voit que les systèmes multi-agents se situent au carrefour de la programmation (ce sont des logiciels), de l'intelligence artificielle (leur autonomie de décision), et des

systèmes répartis (leur décentralisation). Historiquement, on peut replacer les concepts d'agent et de système multi-agent dans l'histoire de l'intelligence artificielle et, de manière duale, dans l'histoire de la programmation.

La notion d'agent rationnel est de fait à la base des débuts de l'intelligence artificielle (IA). Mais cette discipline s'est focalisée sur la modélisation des capacités intelligentes d'une unique entité pour résoudre des problèmes. Il en a résulté une première génération de programmes informatiques évolués, tels les systèmes experts. Mais, même restreint à un domaine spécialisé (domaine expert), un tel agent rationnel système expert était censé résoudre tout seul les problèmes de manière autarcique. L'accent a donc été mis progressivement à partir de la fin des années 1970 sur une résolution distribuée de problèmes, par la coordination d'un certain nombre d'agents rationnels, ce que l'on a alors commencé à appeler des « systèmes multi-agents ». On utilise également le terme quasiment équivalent *d'intelligence artificielle distribuée* (avec son acronyme IAD, en anglais DAI) pour bien montrer l'opposition à l'IA classique, autarcique et centralisée.

Un autre problème des systèmes experts était l'autarcie sociale. Comme ils avaient pour objectif de remplacer les experts humains, ces derniers se retrouvaient « hors de la boucle », avec les conséquences sociales que l'on imagine (désintérêt, déresponsabilisation, inquiétude, etc.). Certains chercheurs ont alors fait évoluer l'objectif initial (concevoir des agents rationnels remplaçant l'homme), vers l'objectif de concevoir des assistants intelligents qui puissent aider l'homme en automatisant et en prenant l'initiative de certaines tâches. Ceci s'est concrétisé en particulier dans le domaine des interfaces homme-machine. Par opposition aux interfaces homme-machine à manipulation directe, l'agent assistant intelligent est ainsi introduit comme une secrétaire virtuelle douée d'initiative, complément d'un logiciel (par exemple, un éditeur de mails) qui apprend vos usages et besoins et peut ainsi les anticiper (par exemple, classer vos messages, automatiser certaines réponses, faire référence à des messages anciens, etc.) [MAE 94].

Si l'on replace maintenant la notion d'agent dans l'histoire de la programmation, on peut voir l'évolution à deux niveaux. Au plan individuel, l'agent se caractérise par une notion d'action persistante [GAS 99], qui tente de manière autonome (et avec initiative) d'accomplir la mission qui lui a été assignée. On sort donc du cadre procédural classique, ce qui permet à l'agent de faire face à des situations non nécessairement prévues à l'avance par le concepteur. On sort également du cadre uniquement déclaratif, puisque l'agent doit pouvoir contrôler lui-même (forme supplémentaire d'autonomie) l'utilisation de ses ressources pour accomplir sa mission. Au plan collectif, on peut aussi considérer l'idée de système multi-agent comme une évolution de la notion de composant logiciel (objet) pour lequel le couplage entre composants est abordé au niveau des connaissances et pas au niveau

des types de données [JEN 99]. Ceci permet donc un couplage sémantiquement de niveau plus élevé et donc moins rigide et plus adaptable.

Tour d'horizon

Le terme d'agent recouvre beaucoup de sens différents selon les communautés et les domaines d'applications et donc les perspectives et techniques mises en avant. Nous introduisons ci-dessous un certain nombre de « domaines » ou « points de vue » sur les agents. Ils correspondent pour certains à différents points de vue historiques qui auront forgé une certaine vision des agents, et pour d'autres à des points de vue applicatifs qui auront amené certaines problématiques plus ou moins spécifiques (par exemple, la mobilité d'agents). Ce paragraphe ne prétend donc pas être une typologie complète, mais se veut plutôt être un tour d'horizon.

Ces différents domaines se recoupent plus ou moins. On peut voir également qu'ils cadrent plus ou moins complètement avec la définition que nous avons donnée plus haut d'un agent, car cela dépend des caractéristiques (autonomie, organisation, mais aussi raisonnement, mobilité, apprentissage, etc.) qui sont mises en avant selon chaque domaine.

Agent cognitif. L'intelligence artificielle classique s'est concentrée très tôt sur l'expression, sur des bases logiques, du comportement délibératif d'un agent rationnel en fonction de ses croyances et de ses buts. Ceci a donné lieu plus tard aux premières architectures modernes d'agents dits *cognitifs* (par exemple Agent0 et BDI, voir au chapitre 2).

Agent logiciel. Les premiers et les plus simples sont les démons Unix (processus informatiques autonomes capables de se réveiller à certaines heures ou en fonction de certaines conditions). Les virus informatiques en sont des versions plus sophistiquées (notamment douées de la capacité de reproduction) et malfaisantes. Notons qu'il est difficile de donner une définition actuelle (consensuelle) du terme d'agent logiciel. Cela sous-entend souvent aussi une encapsulation en agent (« agentification ») de programmes ou de bases de données plus traditionnels, de manière à favoriser leur coopération. En tout cas, ce terme s'oppose au terme d'agent physique, tel un robot.

Agent mobile. La première plate-forme d'agents mobiles est Telescript, née dans la première moitié des années 1990. Elle a depuis donné lieu à de nombreux descendants, extensions du langage Java (Aglets, Voyager, etc.). L'idée est de donner la capacité à un agent logiciel de se déplacer d'une machine à une autre en fonction des données et informations à traiter ainsi localement. La première motivation est généralement la minimisation des communications distantes (il est en

général moins coûteux de faire migrer le code de traitement que les données à traiter, qui peuvent être beaucoup plus volumineuses). Une autre application est celle de l'informatique nomade (l'agent mobile peut continuer ses traitements sur des serveurs pendant la déconnexion de la machine cliente). L'exemple classiquement mis en avant est la recherche de produits (par exemple un CD) au meilleur prix sur des serveurs commerciaux. Les problèmes principaux actuels sont liés à la sécurité et à l'interopérabilité des agents mobiles sur les différentes plates-formes d'accueil.

Agent assistant. Pour dépasser les limitations des interfaces homme-machine à manipulation directe (rigidité, complexité, etc.), les agents assistants (aussi appelés *agents intelligents*), apportent une adaptation au profil de l'utilisateur et une capacité à anticiper leurs besoins (automatiser certaines tâches, rappeler certaines informations utiles, etc.). Ceci peut se transposer dans le domaine du collectif. Prenons l'exemple d'un rendez-vous entre plusieurs personnes, mis en place par la coopération de leurs agents assistants respectifs (pouvant résider sur des agendas électroniques (PDA) ou ordinateurs).

Agent robotique. On peut considérer l'architecture logicielle de contrôle d'un robot comme un agent. Cette architecture peut d'ailleurs être testée et entraînée en simulation avant d'être mise en œuvre dans un vrai robot. Cependant, la réalité physique d'un robot apporte des problèmes spécifiques (imprécision de la perception et de l'action, temps réel, évolution du monde) qui rendent particulièrement difficile, mais aussi particulièrement riche, la conception de tels agents robotiques. On peut aussi envisager des coopérations entre robots, par exemple dans le cadre de la compétition de robots footballeurs (RoboCup), présentée par ses organisateurs comme un nouveau problème étalon de l'intelligence artificielle.

Résolution distribuée de problèmes. Historiquement, le point de départ des systèmes multi-agents est la résolution distribuée (décentralisée) de problèmes (DPS en anglais). L'accent a été mis sur les techniques de coordination (et donc les modèles d'organisations) entre divers agents autonomes amenés à résoudre un problème ensemble (par décomposition du problème, allocation des tâches, échange de plans, détection et résolution d'incohérences, recombinaison des résultats partiels, etc.). De tels modèles de résolution distribuée sont appliqués à des domaines variés, comme le contrôle de processus et le commerce électronique.

Agent réactif. En complément de la résolution distribuée décrite ci-dessus, qui est principalement cognitive et d'approche descendante, s'est développée (en particulier en France, comme le fait remarquer Les Gasser dans sa préface) une approche ascendante de la conception de systèmes multi-agents. Cette approche alternative est souvent basée sur l'émergence de comportements collectifs à partir de comportements individuels relativement simples (réagissant à des stimuli, d'où leur nom d'agents réactifs) et n'ayant pas ou très peu de capacités de représentation et de

raisonnement. Les métaphores issues de sociétés animales (le plus souvent des fourmis) fournissent en particulier des bases de comportements possédant des propriétés collectives recherchées (par exemple, la collecte et le tri pour des robots, le routage adaptatif pour des réseaux).

Vie artificielle. Le domaine de la vie artificielle est né comme alternative à l'intelligence artificielle classique. L'idée est de chercher à modéliser et simuler les propriétés fondamentales de la vie (adaptation, reproduction, etc.) plutôt que des systèmes intelligents (par exemple, un joueur d'échecs ou un démonstrateur de théorèmes) mais très fragiles et inaptes à la vie réelle. On procède à l'importation (et donc détournement) de métaphores issues de la vie réelle et à leur application informatique (par exemple comme techniques d'optimisation ou d'adaptation). Des exemples maintenant classiques sont : les réseaux de neurones artificiels issus de la neurophysiologie, les algorithmes génétiques issus de la théorie de l'évolution et de la génétique, et les algorithmes inspirés de fourmis déjà mentionnés ci-dessus. Certaines approches d'adaptation ou d'évolution des agents proviennent de la vie artificielle.

Simulation multi-agent. Pour certains phénomènes complexes dont on ne connaît pas de modèle global (par exemple, des équations différentielles pour la dynamique des fluides), une alternative possible est de modéliser directement les comportements des entités concernées ainsi que leurs interactions. Ceci peut s'appliquer à des phénomènes physiques (par exemple, des milieux granulaires), sociaux, écosystèmes, etc. Ce domaine est en pleine expansion et est introduit au chapitre 7.

Agent de loisir. Notre société de loisir est à la recherche de compagnons de loisir autonomes et en interaction avec nous. Les jeux vidéo les plus récents intègrent des personnages dont les comportements recherchés sont de plus en plus sophistiqués et donc intéressants pour les joueurs. Le robot chien Aibo de Sony représente une des versions les plus avancées technologiquement d'un tel agent de loisir.

Les systèmes multi-agents comme combinaison de perspectives

Les différents domaines d'agents cités ci-dessus ne sont généralement pas mutuellement exclusifs et peuvent être combinés car ils sont complémentaires. Notons à ce propos, que tous ces domaines partagent la notion d'autonomie, premier enjeu que nous avons mis en avant. Par contre, ils respectent plus ou moins le deuxième enjeu, l'organisation (collective). Par exemple, les agents mobiles restent aujourd'hui encore fondamentalement des solitaires. Mais ils auraient beaucoup à gagner à communiquer, ou mieux encore à coopérer entre eux (par échange d'informations, division et coopération des tâches, etc.). Il existe d'ailleurs un

continuum entre des agents autonomes mais complètement solitaires et des systèmes d'agents complètement organisés, les agents communicants se situant quelque part entre les deux.

De manière plus générale, comme nous l'avons déjà signalé au début de cette introduction, la décentralisation des logiciels devient maintenant inéluctable du fait de la complexité des applications informatiques. La double problématique d'autonomie et d'organisation, qui est à la base des systèmes multi-agents, va donc à notre avis concerner progressivement tous les domaines actuels des agents (et de l'informatique).

Domaines abordés et non abordés dans cet ouvrage

Nous nous concentrerons donc ici sur les « systèmes multi-agents », dans lesquels la double problématique d'autonomie et d'organisation est fondamentale. Nous avons préféré laisser de côté (au moins dans ce premier ouvrage) la description en tant que telle de deux sous-domaines pourtant importants des agents : les agents mobiles et les agents assistants. Ceci car ils sont restés, pour le moment encore, relativement distincts des systèmes multi-agents. Les principaux travaux en matière d'agents mobiles portent sur la mobilité et la sécurité. Les études sur les agents assistants, dont la caractéristique principale est la capacité d'initiative et de conseil, portent sur les techniques d'apprentissage du profil d'utilisation.

Cependant, la problématique générale de l'apprentissage des agents (et non du seul profil d'utilisation) est un sujet important et actuel, en particulier l'apprentissage collectif pour les systèmes multi-agents. Ce sujet devrait d'ailleurs être abordé sous la forme d'un chapitre dans le prochain ouvrage.

Notez également que la notion d'organisation, qui figure pourtant parmi les principes de base des systèmes multi-agents, ne fait pas encore l'objet d'un chapitre distinct dans ce premier ouvrage. Sa nature plus conceptuelle qu'informatique fait qu'elle « diffuse » à de multiples niveaux, en particulier : pour l'aspect statique, on la retrouve dans les représentations mutuelles des compétences et activités entre plusieurs agents, qui sont décrites au chapitre 2 (architectures d'agents) ; pour l'aspect contrôle dynamique, on la retrouve dans les protocoles de coordination des activités, qui sont décrits au chapitre 4 (coordination d'agents cognitifs) ; pour l'aspect méthodologie de conception, on la retrouve dans la définition des différents rôles pour une organisation et spécialisation du travail, qui devrait être abordée sous forme d'un chapitre sur les méthodologies dans le prochain ouvrage.

Plan de l'ouvrage

Le chapitre 1, « Principes généraux et applications », introduit les principes et de premiers exemples d'application des systèmes multi-agents. Le chapitre 2, « Modèles et architectures d'agents », détaille les principes de conception et construction d'un agent à travers ce que l'on appelle une architecture d'agent. Le chapitre 3, « Modèles de communication », présente les moyens d'interaction entre les agents par des langages de communication. Le chapitre 4, « Modèles de coordination d'agents cognitifs », traite des techniques de coordination entre agents pour accomplir des tâches en commun, selon une organisation collective. Le chapitre 5, « Environnements de développement », présente les environnements de développement effectif d'agents, en particulier des plates-formes de construction et d'exécution d'agents. Le chapitre 6, « Application des systèmes multi-agents dans les télécommunications », introduit les diverses applications des systèmes multi-agents au domaine des télécommunications (réseau, recherche d'information et commerce électronique). Le chapitre 7, « Systèmes multi-agents et écosystèmes », introduit le domaine en pleine expansion de la simulation, en particulier celle des écosystèmes.

Bibliographie

- [BRA 97] BRADSHAW J.M. (dir.), *Software Agents*, AAAI Press-MIT Press, 1997.
- [FER 95] FERBER J., *Les Systèmes Multi-Agent – Vers une Intelligence Collective*, InterEditions, 1995.
- [GAS 99] GASSER L., « Agents and Concurrent Objects – Interview of Les Gasser by Jean-Pierre Briot », Dennis Kafura et Jean-Pierre Briot (eds), *IEEE Concurrency*, Special Series on Actors and Agents, vol. 6, n° 4, p. 74-81, octobre-décembre 1998.
- [JEN 99] JENNINGS N.R., « On Agent-Based Software Engineering », *Artificial Intelligence*, n° 117, p. 277-296, 2000.
- [MAE 94] MAES P., « Agents that Reduce Work and Information Overload, Special Issue on Intelligent Agents », Doug Riecken (dir.), *Communications of the ACM*, vol. 37, n° 7, p. 30-40, juillet 1994.

Chapitre 1

Systemes multi-agents : principes généraux et applications

1.1. Des agents autonomes aux systèmes multi-agents

Dans cette section, nous établirons tout d'abord une définition du concept d'agent autonome ainsi que des caractéristiques d'un tel agent. Nous verrons ensuite, à travers un bref historique des travaux effectués dans le domaine, les défis auxquels les chercheurs ont eu à faire face dans le domaine de la technologie agent.

1.1.1. *Qu'est ce qu'un agent ?*

Le concept d'agent a été l'objet d'études depuis plusieurs décennies dans différentes disciplines. Il a été utilisé non seulement dans les systèmes à base de connaissances, la robotique, le langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans des disciplines comme la philosophie et la psychologie. Aujourd'hui, avec l'avènement de nouvelles technologies et l'expansion de l'internet, ce concept est encore associé à de nouvelles applications comme *l'agent ressource*, *l'agent courtier*, *l'assistant personnel*, *l'agent interface*, *l'agent ontologique*, etc. Dans la littérature spécialisée, on trouve une multitude de définitions des agents. Elles se ressemblent toutes, mais diffèrent selon le type d'application pour lequel est conçu l'agent. Nous trouvons dans [FRA 97] une discussion sur les différentes définitions attribués aux agents ainsi que la différence

entre un agent et un programme classique. A titre d'exemple, voici l'une des premières définitions de l'agent due à Ferber [FER 95] : « Un agent est une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle-même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents ».

Il ressort de cette définition des propriétés-clés comme l'*autonomie*, l'*action*, la *perception* et la *communication*. D'autres propriétés peuvent être attribuées aux agents. Nous citons en particulier la *réactivité*, la *rationalité*, l'*engagement* et l'*intention*.

Récemment, Jennings, Sycara et Wooldridge [JEN 98a] ont proposé la définition suivante pour un agent : « Un agent est un système informatique, *situé* dans un environnement, et qui agit d'une façon *autonome* et *flexible* pour atteindre les objectifs pour lesquels il a été conçu ».

Les mots « situé », « autonomie » et « flexible » sont définis comme suit :

- *situé* : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement (systèmes de contrôle de processus, systèmes embarqués, etc.) ;
- *autonome* : l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne ;
- *flexible* : l'agent dans ce cas est :
 - *capable de répondre à temps* : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans les temps requis,
 - *proactif* : l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment,
 - *social* : l'agent doit être capable d'interagir avec les autres agents (logiciels et humains) quand la situation l'exige afin d'accomplir ses tâches ou aider ces agents à accomplir les leurs.

Cette définition est conforme à notre vision de l'entité agent et, par conséquent, nous avons convenu de l'adopter tout au long de ce chapitre. Nous nous référons donc à des agents cognitifs capables de planifier et de prendre des décisions et non à des agents réactifs comme le ferait l'écologie [BOU 01b].

Bien entendu, selon les applications, certaines propriétés sont plus importantes que d'autres. Il peut même s'avérer que pour certains types d'applications, des propriétés additionnelles soient requises. Cependant, la présence des propriétés que l'on vient de voir comme l'autonomie, la flexibilité et la sociabilité donne naissance

au paradigme agent tout en le distinguant des systèmes conventionnels comme les systèmes distribués, les systèmes orientés objets et les systèmes experts.

Voyons maintenant comment l'idée d'agent que nous venons de définir a évolué au cours des dernières années.

1.1.1.1. *Historique*

La majeure partie des composantes d'un agent autonome fait appel à des techniques de l'intelligence artificielle (IA). On pourrait donc croire que la construction d'un agent autonome est au centre des problèmes étudiés par l'intelligence artificielle. Mais ce n'est pas le cas. En effet, avant les années 1980, la recherche en intelligence artificielle s'est plutôt concentrée sur les diverses composantes d'un agent, en supposant que l'on pourrait un jour assembler ces différentes parties afin de construire un agent de manière assez directe. Durant cette période, le champ d'intérêt qui a le plus en commun avec les agents autonomes est celui de la planification

1.1.1.1.1. La planification en intelligence artificielle

La planification [ALL 90, CHA 86, NEW 61, WIL 88] est le sous-domaine de l'intelligence artificielle qui cherche à répondre à la question : « Que doit-on faire ? », c'est-à-dire quelles actions poser et dans quel ordre ? Comme un agent est une entité qui pose des actions, on voit rapidement l'intérêt de la planification pour les agents autonomes. Un des premiers systèmes de planification fut le système STRIPS (*STanford Research Institute Problem Solver*, la distinction entre planificateur et solveur n'étant pas encore claire à l'époque) [FIK 71]. Dans un système STRIPS typique, on retrouve habituellement les composantes suivantes :

- un modèle symbolique de l'environnement de l'agent, exprimé typiquement dans un sous-ensemble limité de la logique du premier ordre ;
- une spécification symbolique des actions que l'agent peut accomplir, généralement sous forme de *préconditions*, *actions*, *effets*, qui représentent l'état de l'environnement avant que l'action soit posée ainsi que les conséquences de cette action ;
- un algorithme de planification, qui est apte à manipuler les symboles définis et qui génère, en sortie, un plan représentant les actions que devra poser l'agent pour atteindre son but.

Dans les années 1970 et jusqu'au début des années 1980, la recherche sur la planification s'est donc surtout orientée sur les trois aspects exposés ci-dessus. Principalement, on voulait démontrer l'efficacité des algorithmes de planification développés. Malheureusement, on s'est vite aperçu que, bien qu'ils donnent de bons résultats pour des petits problèmes (par exemple, le monde des blocs), les

algorithmes de planification ont une performance plutôt médiocre lorsqu'on les applique à des problèmes du monde réel. En effet, comme la taille de l'espace de recherche des algorithmes de planification croît de façon exponentielle avec la complexité de la tâche pour laquelle on cherche à établir un plan, et que les tâches du monde réel sont en général très complexes, on obtient par conséquent un espace de recherche extrêmement grand, ce qui donne des temps de recherche proportionnellement longs.

Devant l'apparence d'échec des chercheurs en IA à développer des algorithmes de planification efficaces dans le monde réel, certains chercheurs ont commencé à questionner les suppositions qui sont à la base de l'intelligence artificielle symbolique. Plus particulièrement, on émit de grandes réserves quant à l'approche logicienne¹ utilisée jusqu'à maintenant.

1.1.1.1.2. Agents réactifs

Un des grands noms parmi les critiques du raisonnement symbolique fut Brooks, qui, par le biais de plusieurs articles [BRO 86, BRO 91a, BRO 91b], exposa son opposition au modèle symbolique et proposa une approche alternative que l'on appelle aujourd'hui IA réactive. Selon lui, le comportement intelligent devrait émerger de l'interaction entre divers comportements plus simples. Ainsi, au sein de son programme de recherche, il a développé l'architecture *subsumption*. Dans cette architecture, on construit des agents sans utiliser de représentation symbolique ni de raisonnement. Un agent est alors constitué d'un ensemble de comportements permettant d'accomplir une tâche donnée. Chaque comportement est une machine à états finis qui établit une relation entre une entrée sensorielle et une action en sortie.

Typiquement, l'ensemble des comportements est représenté sous la forme d'une hiérarchie dans laquelle les couches des niveaux inférieurs représentent des comportements moins abstraits, et les couches des niveaux supérieurs, des comportements plus abstraits. Le développement d'un agent devient donc un processus dans lequel on devra expérimenter les nouveaux comportements. Ceci est habituellement accompli en plaçant l'agent dans son environnement et en observant les résultats.

Bousquet et Lepage [BOU 01b] reviennent en détail sur ce type d'approche, généralement utilisée pour la simulation. Malgré la simplicité apparente et les bons résultats obtenus pour certaines applications [BRO 91a, CHA 86, KEN 96, WAV 95], bien des reproches ont été adressés à cette approche dite « réactive », parmi lesquels, il convient de voir que :

1. Une approche selon laquelle un agent manipule des prédicats de la logique du premier ordre pour effectuer son raisonnement.

- si les agents ne possèdent pas de modèle de leur environnement, ils doivent posséder suffisamment d'informations locales leur permettant de choisir une action acceptable ;
- comme les agents basent leurs décisions sur des informations locales, il est difficile de voir comment ils pourraient tenir compte des informations non locales ;
- il est difficile de voir comment un agent purement réactif peut apprendre de son expérience et améliorer ainsi ses performances ;
- le comportement global d'un agent *devrait émerger* des interactions entre les divers comportements qui le composent ; cette *émergence* rend donc très difficile la tâche de construire un agent pour effectuer une tâche spécifique ;
- s'il est assez simple de bâtir un agent qui comporte très peu de couches, l'exercice devient beaucoup plus compliqué lorsque l'on a besoin de plusieurs couches. Les interactions dynamiques entre les diverses couches deviennent trop complexes à comprendre.

1.1.1.1.3. Architectures hybrides

Dès le début des années 1990, on savait que les systèmes réactifs pouvaient bien convenir pour certains types de problèmes et moins bien pour d'autres. De même, pour la plupart des problèmes, les solutions de l'IA classique, basées uniquement sur la planification, ne conviennent pas non plus. On commence dès lors à explorer la possibilité de combiner les deux approches afin d'obtenir une architecture hybride [CHA 94a, CHA 94b, CHA 96b, CHA 96c, FER 92, FIS 99, GEO 87]. Dans ce cas, un agent est composé de plusieurs couches, arrangées selon une hiérarchie, la plupart des architectures considérant que trois couches suffisent amplement. Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision qui se situe au niveau des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement, c'est-à-dire du raisonnement tenant compte des autres agents.

1.1.1.1.4. Architectures BDI

Dans le cadre du raisonnement pratique, un raisonnement orienté vers la prise en compte des états mentaux, les chercheurs ont développé l'architecture BDI [BRA 87, BRA 88, GEO 87, RAO 91a, RAO 91b, RAO 92, SIN 94] (de l'anglais *Belief, Desire, Intention* pour croyance, désir et intention), une architecture basée sur le *raisonnement pratique*². Dans ce type d'architectures, les agents sont

2. Traduction du terme anglais *practical reasoning*, c'est-à-dire, le raisonnement « pratique » tel qu'il se pratique chez les humains.

généralement représentés par un « état mental » ayant les attitudes mentales suivantes :

- *les croyances* : ce que l'agent connaît de son environnement ;
- *les désirs* : les états possibles envers lesquels l'agent peut vouloir s'engager ;
- *les intentions* : les états envers lesquels l'agent s'est engagé, et envers lesquels il a engagé des ressources.

Un agent BDI doit donc mettre à jour ses croyances avec les informations qui lui proviennent de son environnement, identifier les options qui lui sont offertes, filtrer ces options afin de déterminer de nouvelles intentions et poser ses actions au vu de ses intentions.

1.1.1.2. *Les applications des agents autonomes*

Les programmes actuels interagissent avec leur usager en mode dit de *manipulation directe*. C'est-à-dire que le programme fait exactement ce que l'usager lui demande de faire. Cependant, il pourrait être intéressant d'avoir des programmes qui prennent l'initiative dans certaines circonstances. Ainsi, un logiciel prendrait une part active pour aider l'usager à accomplir ses tâches [MAE 94a, MAE 94b]. Des applications de tels agents se retrouvent principalement dans les lecteurs de courriels, les lecteurs de groupes de discussions actifs ainsi que les navigateurs actifs.

A part les agents interfaces, la technologie agent est mise en œuvre dans des applications utilisant des agents mobiles [GRA 98, RUS 96] ou des agents émotionnels [BAT 92, BAT 94, HAY 98]. Le premier type d'agent est capable de migrer d'une machine à une autre pour exécuter le programme dont il est porteur afin de, par exemple, ne pas surcharger le réseau. Le deuxième, quant à lui, a été conçu afin d'« exprimer » des « émotions » comme le feraient des humains placés dans la même situation. Ce type d'agent s'est déjà avéré utile dans des applications du type animation par exemple [HAY 98].

En fait, l'agent en tant qu'entité individuelle est d'un usage limité dans bien des cas, surtout pour des applications distribuées actuelles dans lesquelles un ensemble d'agents doit mettre en commun des compétences et des connaissances. Ce type d'agents forment ce que l'on appelle des systèmes multi-agents.

1.1.2. Les systèmes multi-agents

1.1.2.1. Présentation

Un système multi-agent est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissant, le plus souvent, selon des modes de *coopération*, de *concurrence* ou de *coexistence* [CHA 94a, CHA 96b, MOU 96].

Un SMA est généralement caractérisé ainsi :

- chaque agent a des informations ou des capacités de résolution de problèmes limitées (ainsi, chaque agent a un point de vue partiel),
- il n'y a aucun contrôle global du système multi-agent,
- les données sont décentralisées,
- le calcul est asynchrone.

Les SMA sont particulièrement appropriés pour représenter des problèmes qui peuvent être abordés selon de multiples méthodes de résolution, de multiples perspectives et/ou de multiples solveurs. Ces systèmes possèdent les avantages traditionnels de la résolution distribuée et concurrente de problèmes comme la modularité, la vitesse (grâce au parallélisme), et la fiabilité (grâce à la redondance). Ils héritent aussi des bénéfices envisageables de l'intelligence artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité, mais surtout ils ont l'avantage de faire intervenir des schémas d'interaction sophistiqués. Les types courants d'interaction incluent la coopération (travailler ensemble à la résolution d'un but commun), la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) et la négociation (parvenir à un accord acceptable pour toutes les parties concernées).

Bien que les SMA offrent de nombreux avantages potentiels, ils doivent aussi relever beaucoup de défis. Voici les problèmes inhérents à la conception et à l'implémentation des SMA, d'après [BON 88] :

- Comment formuler, décrire, décomposer, allouer les problèmes et synthétiser les résultats ?
- Comment permettre aux agents de communiquer et d'interagir ? Quoi et quand communiquer ?
- Comment assurer que les agents agissent de manière cohérente i) en prenant leurs décisions ou actions, ii) en gérant les effets non locaux de leurs décisions locales et iii) en évitant les interactions nuisibles ?

- Comment permettre aux agents individuels de représenter et raisonner sur les actions, plans et connaissances des autres agents afin de se coordonner avec eux ?
- Comment raisonner sur l'état de leurs processus coordonnés (comme l'initialisation ou la terminaison) ?
- Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions ?
- Comment trouver le meilleur compromis entre le traitement local au niveau d'un seul agent et le traitement distribué entre plusieurs agents (traitement distribué qui induit la communication) ? Plus généralement, comment gérer la répartition des ressources limitées ?
- Comment éviter ou amoindrir un comportement nuisible du système global, comme les comportements chaotiques ou oscillatoires ?
- Comment concevoir les plates-formes technologiques et les méthodologies de développement pour les SMA ?

Les SMA sont à l'intersection de plusieurs domaines scientifiques : informatique répartie et génie logiciel, intelligence artificielle, vie artificielle. Ils s'inspirent également d'études issues d'autres disciplines connexes, notamment la sociologie, la psychologie sociale, les sciences cognitives et bien d'autres. C'est ainsi qu'on les trouve parfois à la base des :

- systèmes distribués [FAG 95, HAL 95] ;
- interfaces hommes-machines [KOB 89, NEG 95] ;
- bases de données et bases de connaissances distribuées coopératives [BAB 97] ;
- systèmes pour la compréhension du langage naturel [ALL 80, APP 85, COH 79, COH 90a, COH 90b, GAL 88, KAP 98, LIT 87] ;
- protocoles de communication et réseaux de télécommunication [BOU 98, HAL 95, MAL 98, NWA 99] ;
- programmation orientée agents et génie logiciel [JEN 01, SHO 93, THO 93] ;
- robotique cognitive et coopération entre robots [LAK 99, LES 91a, LES 94] ;
- applications distribuées comme le *web*, l'internet, le contrôle du trafic routier, le contrôle aérien, les réseaux d'énergie, etc. [CHA 96a, JEN 98b].

Les recherches dans le domaine des systèmes multi-agents poursuivent deux objectifs majeurs : le premier concerne l'analyse théorique et expérimentale des mécanismes qui ont lieu lorsque plusieurs entités autonomes interagissent. Le second s'intéresse à la réalisation de programmes distribués capables d'accomplir des tâches complexes grâce à la coopération et à l'interaction. Leur position est donc double : d'un côté, elles se placent au sein des sciences cognitives, des sciences sociales et des sciences naturelles pour à la fois modéliser, expliquer et simuler des

phénomènes naturels, et susciter des modèles d'auto-organisation ; d'un autre côté, elles se présentent comme une pratique, une technique qui vise à la réalisation de systèmes informatiques complexes à partir des notions d'agent, de communication, de coopération et de coordination d'actions.

Les agents autonomes et les systèmes multi-agents représentent une nouvelle approche pour l'analyse, la conception et l'implantation des systèmes informatiques complexes. La vision basée sur l'entité agent offre un puissant répertoire d'outils, de techniques et de métaphores qui ont le potentiel d'améliorer considérablement les systèmes logiciels.

En accord avec Jennings [JEN 01], nous pensons que la technologie agent va représenter dans les prochaines années un nouveau paradigme de programmation pour le génie logiciel. Un nouveau paradigme similaire à la *programmation orientée objet* et qui d'ailleurs pourrait s'appeler la *programmation orientée agent*.

A ce propos, il convient de ne pas confondre « agent » et « objet ». Tout d'abord, comme les agents, les objets encapsulent leur état interne (leurs données). Ils peuvent également poser des actions sur cet état par le biais de leurs méthodes et ils communiquent en s'envoyant des messages. A ce niveau, ils diffèrent des agents par leur degré d'autonomie. En effet, une méthode doit être invoquée par un autre objet pour pouvoir être déclenchée. Un agent, quant à lui, recevra une requête et décidera de son propre gré s'il doit poser ou non une action. Une seconde différence provient du caractère flexible (réactif, proactif et social) du comportement d'un agent. Même si certains diront qu'il est possible de bâtir un programme orienté objet intégrant ces caractéristiques, on doit également voir que le modèle standard d'un objet ne dit rien à propos de ces types de comportements. Une troisième différence provient du fait que l'on considère un agent comme étant lui-même une source de contrôle au sein du système tandis que, dans un système orienté objet, on n'a qu'une seule source de contrôle.

Nous dressons maintenant un bref historique des travaux sur les SMA, en présentant des travaux précurseurs de ces systèmes ainsi que quelques exemples des premières applications développées.

1.1.2.2. Historique

En 1980, un groupe de chercheurs s'est réuni pour discuter des défis concernant la résolution « intelligente » de problèmes dans un système comportant plusieurs solveurs de problèmes. Lors de cette réunion, il a été décidé que l'intelligence artificielle distribuée n'axerait pas ses travaux sur les détails de bas niveau de la parallélisation ni sur la manière de paralléliser les algorithmes centralisés, mais plutôt sur le fait de savoir comment un groupe de solveurs de problèmes pourrait coordonner ses efforts afin de résoudre des problèmes efficacement.

On peut dire que les SMA ont vu le jour avec l'avènement de l'intelligence artificielle distribuée (IAD). A ses débuts toutefois, l'IAD ne s'intéressait qu'à la coopération entre solveurs de problèmes. On divisait en général un problème en sous-problèmes et l'on allouait ces sous-problèmes à différents solveurs qui devaient coopérer pour élaborer des solutions partielles. Celles-ci sont finalement synthétisées en une réponse globale au problème initial. Ainsi, l'IAD, au départ, privilégiait le problème à résoudre tout en mettant l'accent sur la résolution d'un tel problème par de multiples entités intelligentes. Dans les SMA d'aujourd'hui, les agents sont (entre autres) autonomes, possiblement préexistants et généralement hétérogènes. Dans ce cas, l'accent est plutôt mis sur le fait de savoir comment les agents vont coordonner leurs connaissances, buts et plans pour agir et résoudre des problèmes.

1.1.2.2.1. Le modèle des acteurs

Un des premiers modèles proposés à l'époque de l'IAD fut le modèle des acteurs [AGH 86, AGH 88]. Les acteurs sont des composantes autonomes d'un système qui communiquent par échanges de messages asynchrones. Ils sont composés d'un ensemble de primitives parmi lesquelles on retrouve :

- *Create* : pour créer un acteur à partir d'un ensemble de paramètres décrivant son comportement ;
- *Send* : pour envoyer un message à un autre acteur ;
- *Become* : pour changer l'état local d'un acteur.

Les acteurs s'avèrent être un modèle assez naturel pour le calcul parallèle. Cependant, les divers modèles d'acteurs, comme bien d'autres modèles de l'intelligence artificielle distribuée, font face à un problème de cohérence. Leur granularité fine pose des problèmes de comportement dans des systèmes qui renferment plusieurs acteurs. Ils éprouvent également des difficultés à atteindre des buts globaux avec seulement des connaissances locales.

1.1.2.2.2. Le protocole *Contract Net*

Le protocole du *Contract Net* [DAV 83] fut une des premières solutions au problème d'allocation de tâches auquel doit faire face un ensemble de solveurs de problèmes. Dans ce protocole, les agents peuvent prendre deux rôles : *gestionnaire* ou *contractant*. L'agent qui doit exécuter une tâche donnée (le *gestionnaire*) commence tout d'abord par décomposer cette tâche en plusieurs sous-tâches. Il doit ensuite annoncer les différentes sous-tâches au reste des agents de l'environnement. Les agents qui reçoivent une annonce de tâche à accomplir peuvent ensuite faire une proposition devant refléter leur capacité à remplir cette tâche. Le gestionnaire rassemble ensuite toutes les propositions qu'il a reçues et alloue la tâche à l'agent ayant fait la meilleure proposition.

1.1.2.2.3. Les premières applications

Parmi les premières applications développées à l'aide de la technologie multi-agent, on retrouve une application dans le contrôle du trafic aérien et une autre dans la surveillance de véhicules motorisés. Pour le contrôle du trafic aérien, Cammarata [CAM 88] a étudié des stratégies de coopération pour pouvoir résoudre les conflits entre les plans d'un groupe d'agents. Ces stratégies ont ensuite pu être appliquées dans le cadre d'un système de contrôle du trafic aérien. Dans ce système, chaque agent (représentant un avion) cherche à bâtir un plan de vol qui devrait le garder à une distance sécuritaire de chacun des autres agents. Dans le cas où des agents se retrouveraient dans une situation conflictuelle, ils doivent choisir parmi eux un agent apte à élaborer un nouveau plan de vol sans engendrer de nouveaux conflits. Pour faire ce choix, Cammarata et ses collègues ont opté pour deux stratégies qu'ils ont comparées entre elles. La première a consisté à choisir l'agent le mieux informé ou l'agent le moins contraint pour jouer le rôle de planificateur central et élaborer un nouveau plan de vol qui résoudrait le conflit. La deuxième a consisté à faire le partage suivant : il revient à l'agent le mieux informé d'élaborer un nouveau plan de vol et à l'agent le moins contraint de l'exécuter.

La surveillance de véhicules motorisés constituait le domaine d'application du système DVMT (*Distributed Vehicle Monitoring Task*) [COR 83]. Ce système avait comme tâche principale d'identifier quel type de véhicule circulait dans la zone où étaient placés ses senseurs. À partir de ces interprétations, il devait tenter d'établir une description des mouvements des véhicules dans la région qu'il supervisait. Comme bien d'autres systèmes multi-agents de l'époque, le système DVMT utilisait un *blackboard* pour coordonner les efforts de ses différents agents. Un *blackboard* [ENG 89] est simplement une structure de données partagées entre divers agents. Ces derniers peuvent la consulter pour obtenir des informations sur l'état actuel du problème ou y écrire la partie de la solution qu'ils ont obtenue.

1.2. Interactions et coopération entre agents

Un système multi-agent se distingue d'une collection d'agents indépendants par le fait que les agents interagissent en vue de réaliser conjointement une tâche ou d'atteindre conjointement un but particulier. Les agents peuvent interagir en communiquant directement entre eux, par l'intermédiaire d'un autre agent ou même en agissant sur leur environnement. Chaque agent peut être caractérisé par trois dimensions : ses buts, ses capacités à réaliser certaines tâches et les ressources dont il dispose. Les interactions des agents d'un SMA sont motivées par l'interdépendance des agents selon ces trois dimensions : leurs buts peuvent être compatibles ou non ; les agents peuvent désirer des ressources que les autres possèdent ; un agent X peut disposer d'une capacité nécessaire à un agent Y pour l'accomplissement d'un des plans d'action de Y.

Dans le cadre des systèmes multi-agents, on doit traiter la question fondamentale qui est l'implantation de la coopération entre les agents [DEM 91]. Durfee et ses collègues [DUR 89] ont proposé quatre objectifs génériques pour établir la coopération dans un groupe d'agents : augmenter le taux de finalisation des tâches grâce au parallélisme, augmenter le nombre de tâches réalisables grâce au partage de ressources (information, expertise, dispositifs physiques, etc.), augmenter les chances de finaliser des tâches en les dupliquant et en utilisant éventuellement des modes de réalisation différents, diminuer les interférences entre tâches en évitant les interactions négatives.

En fait, on peut caractériser un système par le type de coopération mis en œuvre qui peut aller de la coopération totale à l'antagonisme total [MOU 96]. Des agents totalement coopératifs peuvent changer leurs buts pour répondre aux besoins des autres agents afin d'assurer une meilleure coordination entre eux. Il peut en résulter des coûts de communication élevés.

Les agents antagonistes, par contre, ne vont pas coopérer et, dans ce cas, leurs buts respectifs se trouvent bloqués. Dans de tels systèmes, les coûts de communication sont minimaux. La plupart des systèmes réels se situe entre les deux extrêmes : coopération totale et antagonisme total. La coopération totale est mise en œuvre par exemple dans les approches de résolution coopérative de problèmes distribués (*Cooperative Distributed Problem Solving* ou CDPS) proposée par Durfee et ses collègues [DUR 89]. Dans ces approches, les agents coopèrent pour résoudre des problèmes qu'ils ne peuvent pas résoudre individuellement. Tout d'abord chaque agent utilise ses connaissances et ressources pour résoudre localement un ou plusieurs sous-problèmes. Les solutions partielles à tous les sous-problèmes sont par la suite intégrées. C'est par exemple le cas des avions qui volent dans un même espace aérien et qui, de ce fait, ont différentes perceptions de leur espace environnant obtenues par leurs senseurs respectifs. Ce n'est qu'en combinant ces différentes perceptions que ces avions peuvent obtenir une vue globale afin de résoudre les conflits potentiels qui peuvent se produire entre eux.

Ferber [FER 95] remarque que les chercheurs ont développé différents points de vue sur la coopération. Ainsi, on peut considérer la coopération comme une attitude adoptée par les agents qui décident de travailler ensemble ou on peut adopter le point de vue d'un observateur extérieur au système multi-agent qui interprète *a posteriori* les comportements des agents pour les qualifier de coopératifs ou non selon des critères préétablis tels que l'interdépendance des actions ou le nombre de communications effectuées. Dans le cas de la coopération vue comme une attitude intentionnelle, les agents s'engagent dans une action après avoir identifié et adopté un but commun. C'est par exemple le cas d'un groupe de personnes acceptant de travailler dans une entreprise. Des chercheurs comme Galliers [GAL 88] et Conte

[CON 91b] ont souligné l'importance de l'adoption d'un but commun pour ce type de coopération, qu'ils considèrent comme un élément essentiel de l'activité sociale.

1.2.1. *Interactions entre agents coopératifs*

Au fur et à mesure que croît l'intérêt pour les applications composées d'agents s'entraidant pour atteindre un but commun, et que de plus en plus d'agents sont construits pour coopérer avec d'autres agents, il devient important de bien comprendre les principes qui sous-tendent la coopération.

La planification pour un agent individuel, telle que discutée au paragraphe 1.1.1, ne fait que construire une séquence d'actions en ne considérant que les buts de l'agent, ses capacités, et les contraintes imposées par son environnement. Par contre, dans un environnement multi-agent, on doit tenir compte des contraintes que les actions des autres agents placent sur le choix des actions de l'agent, des contraintes que les engagements de l'agent envers les autres imposent sur son propre choix d'action, ainsi que l'évolution imprévisible de l'environnement due à des agents qui ne sont pas modélisés.

Les premiers travaux en intelligence artificielle distribuée étaient principalement concentrés sur des groupes d'agents qui poursuivent des buts communs. On choisit alors de privilégier une approche selon laquelle on planifie avant d'agir. Ainsi, afin de produire un plan cohérent, les agents doivent être en mesure de reconnaître les interactions entre les différents sous-buts pour pouvoir, soit les ignorer, soit les résoudre.

Une autre approche intéressante pour résoudre les interdépendances entre les sous-buts a été le *Functionally Accurate Model* [LES 80, LES 91b] dans lequel les agents n'ont pas besoin d'avoir toutes les informations nécessaires pour résoudre les conflits localement. Les interactions se font plutôt par un échange asynchrone de résultats partiels. A partir de ce modèle sont apparus plusieurs principes de coordination, parmi lesquels on retrouve le *Partial Global Planning* [DUR 87, DUR 88].

La planification partielle globale (en anglais GPP pour *Global and Partial Planning*) est une approche flexible qui permet aux divers agents d'un système de se coordonner dynamiquement. Les agents interagissent en se communiquant leurs plans et leurs buts selon un niveau d'abstraction approprié. Ces communications permettent à chacun d'anticiper les actions futures d'un ou de plusieurs autres agents, augmentant ainsi la cohérence de l'ensemble. Comme les agents coopèrent, le receveur d'un message peut utiliser les informations reçues afin d'ajuster sa propre planification.

Un autre courant de recherche sur la coordination des agents consiste à modéliser explicitement le travail d'équipe qu'effectuent les agents. Cette approche s'avère très pratique dans des environnements dynamiques où les membres de l'équipe peuvent échouer dans leur tâche ou découvrir de nouvelles opportunités. Dans de telles situations, l'équipe doit pouvoir évaluer sa performance et être en mesure de se réorganiser en conséquence. Dans cet ordre d'idées, on parle alors du modèle des *intentions conjointes* [JEN 95, LEV 90]. Un tel modèle est d'ailleurs une extension naturelle du raisonnement pratique (voir paragraphe 1.1.1). Dans ce modèle, on cherche à modéliser l'état mental d'une équipe en action. L'équipe aura une intention conjointe si chaque membre de l'équipe est engagé à compléter l'action de l'équipe. Préalablement à cette intention conjointe, tous les membres de l'équipe auront établi les croyances et les engagements communs appropriés, et ce en s'échangeant des actes (du langage naturel) de requête et de confirmation.

1.2.2. *Interactions entre agents égocentrés*

Les interactions entre agents égocentrés sont basées principalement sur la négociation. Cette dernière devient donc une méthode de coordination et de résolution de conflits. Elle a également été utilisée comme métaphore pour la communication de changements dans les plans, l'allocation de tâches ainsi que la résolution centralisée de violations de contraintes. On peut donc voir que la définition de la négociation est aussi imprécise que celle du concept d'agent. On arrive malgré tout à identifier certaines caractéristiques importantes de la négociation : la présence d'une forme de conflit qui doit être résolu de façon décentralisée par des agents égocentrés ayant une rationalité limitée et des informations incomplètes.

De plus, les agents communiquent et échangent de façon itérative des propositions et des contre-propositions.

Le système PERSUADER de Sycara [SYC 88, SYC 89, SYC 90a, SYC 90b] et les travaux de Rosenschein [ROS 85b, ROS 94] sont les premiers travaux de l'intelligence artificielle distribuée sur la négociation entre agents égocentrés. Le système PERSUADER puise son modèle dans les négociations de conventions collectives. Ces négociations impliquent trois agents (un syndicat, une compagnie et un médiateur) et s'inspirent des négociations entre êtres humains. On y modélise les multiples itérations pendant lesquelles les parties impliquées échangent offres et contre-offres, l'utilité des agents étant multidimensionnelle et privée. Les croyances des agents sont révisées de sorte que l'on peut atteindre une entente grâce à une argumentation persuasive.

Rosenschein, quant à lui, base son travail sur la théorie des jeux. L'utilité est le seul paramètre considéré par les agents et l'on suppose que les agents sont omniscients. Les valeurs de l'utilité des différentes alternatives sont représentées par une matrice des gains qui est commune aux parties impliquées dans la négociation. Chaque partie choisira donc l'alternative qui maximisera son utilité. Malgré l'élégance mathématique de la théorie des jeux, les négociations modélisées par Rosenschein s'appliquent mal à des problèmes du monde réel où :

- les négociations se déroulent dans une certaine incertitude ;
- les négociations impliquent des critères beaucoup plus complexes qu'une utilité à une seule dimension ;
- les utilités des agents sont secrètes ;
- les agents ne sont pas omniscients.

Krauss [KRA 91, KRA 95] de son côté s'est penchée sur le rôle du temps dans les négociations. Cette étude lui a permis de voir que l'importance qu'un agent accorde au temps peut affecter l'entente finale entre les parties.

Comme le commerce électronique devient de plus en plus une réalité, le besoin de techniques de négociation qui tiennent compte des complexités du monde réel grandit. On se doit donc de tenir compte des informations incomplètes, des problèmes de négociations multiples, des échéances pour les négociations et la possibilité de briser des contrats. On a donc ajouté la possibilité d'établir des contrats moins rigides au sein du protocole *Contract Net*, des pénalités étant attribuées lorsqu'un contrat n'est pas rempli complètement [SAN 95].

1.3. Coordination entre agents

De nombreux exemples de coordination existent dans la vie quotidienne : deux déménageurs déplaçant un meuble lourd, deux jongleurs échangeant des balles, des personnes qui parlent à tour de rôle en se passant un micro, etc. Malone [MAL 90] note que deux des composantes fondamentales de la coordination entre agents sont l'allocation de ressources rares et la communication de résultats intermédiaires. Dans ce contexte, les agents doivent être capables de communiquer entre eux de façon à pouvoir échanger les résultats intermédiaires. Pour l'allocation des ressources partagées, les agents doivent être capables de faire des transferts de ressources. Ceci peut d'ailleurs imposer certains comportements à des agents particuliers. Malone conclut qu'il peut être utile de distinguer les liens de contrôle comme une catégorie spéciale de liens de communication par lesquels certains agents transmettent des instructions que d'autres vont être motivés à suivre.

En étudiant les communautés humaines, Mintzberg [MIN 79] a identifié trois processus fondamentaux de coordination : ajustement mutuel, supervision directe et coordination par standardisation. L'*ajustement mutuel* est la forme de coordination la plus simple puisqu'elle se produit quand deux ou plusieurs agents s'accordent pour partager des ressources en vue d'atteindre un but commun. Habituellement, les agents doivent échanger de nombreuses informations et faire plusieurs ajustements de leurs propres comportements en tenant compte de ceux des autres agents. Dans cette forme de coordination, aucun agent n'a un contrôle sur les autres agents et le processus de décision est conjoint. La coordination dans les groupes de pairs (*peer groups*) et dans les marchés (*market*) est habituellement une forme d'ajustement mutuel. La *supervision directe* apparaît quand un ou plusieurs agents ont déjà établi une relation dans laquelle un des agents a un contrôle sur les autres. Cette relation est habituellement établie par ajustement mutuel, comme par exemple dans le cas d'un employé ou d'un sous-contractant qui accepte de suivre les instructions du superviseur. Dans cette forme de coordination, l'agent superviseur contrôle l'utilisation des ressources partagées (par exemple : les ressources humaines, le temps de calcul ou l'argent) par les agents subordonnés. Il peut aussi imposer certains comportements. Dans les cas de *coordination par standardisation* le superviseur coordonne les activités en établissant des procédures que doivent suivre les subordonnés dans des situations identifiées. On trouve de telles procédures dans les entreprises, mais aussi dans les systèmes informatiques.

Malone [MAL 90] suggère qu'en utilisant des processus de coordination fondamentaux proches de ceux de Mintzberg, il est possible de construire des systèmes de coordination sophistiqués dont les plus répandus sont les hiérarchies et les marchés basés respectivement sur les processus de supervision directe et d'ajustement mutuel. Le travail en petit groupe se prête bien à l'ajustement mutuel. Par contre, dès que la taille du groupe grandit et que le nombre de tâches augmente, le nombre de liens et la quantité d'informations échangées peuvent devenir rapidement un handicap sérieux. Un groupe important peut être partagé efficacement en sous-groupes de manière à ce que la plupart des interactions s'effectuent dans les sous-groupes et que les quelques interactions nécessaires entre les sous-groupes soient prises en charge par les superviseurs de ces sous-groupes. Les sous-groupes peuvent être coordonnés par contrôle hiérarchique ou par ajustement mutuel selon les caractéristiques du domaine d'application et des tâches à accomplir. Les marchés sont aussi considérés comme une activité d'organisation de groupe basée sur l'ajustement mutuel dans laquelle chaque agent contrôle des ressources rares (par exemple : la main-d'œuvre, les matières premières, les marchandises et l'argent). Les agents s'accordent pour partager leurs ressources respectives afin d'atteindre un but commun. Les ressources échangées ont un prix explicite ou implicite. Lorsqu'un contrat est conclu, il y a un accord pour que l'agent contracteur devienne le superviseur de l'agent contractant. De nombreuses organisations mettent en œuvre des processus de coordination mixtes basées sur l'ajustement mutuel, la supervision

directe et la standardisation. Ces cadres organisationnels peuvent être des sources d'inspiration lorsque l'on doit établir une structure organisationnelle pour un SMA.

Bien entendu, la coordination est une question centrale pour les SMA et la résolution de problèmes distribués. En effet, sans coordination un groupe d'agents peut dégénérer rapidement en une collection chaotique d'individus. On pourrait penser que la façon la plus simple de s'assurer du comportement cohérent du groupe d'agents serait de le faire par le biais d'un agent centralisateur qui détiendrait des informations de haut niveau sur ces agents. Ainsi, l'agent centralisateur pourrait créer des plans d'action et assigner les tâches aux divers agents du groupe. Cette approche est pratiquement impossible à mettre en œuvre dans des applications réalistes en raison de la difficulté de réaliser un tel agent centralisateur qui puisse tenir compte des buts, des connaissances et des activités de chaque agent : la charge en communication serait énorme, sans compter que l'on perdrait les avantages d'un SMA composé d'agents autonomes [CON 91a]. Le contrôle et les informations doivent alors être distribués parmi les agents.

A cette fin, Jennings [JEN 95] propose un modèle intéressant qui se base sur l'hypothèse de la centralité des engagements et des conventions. Les engagements sont vus comme des promesses en vue de réaliser certaines actions, alors que les conventions constituent le moyen de faire le suivi de ces engagements dans des circonstances changeantes. Grâce aux engagements des autres agents, un agent peut prédire les actions qu'il est susceptible d'effectuer et déterminer ainsi les interdépendances avec ses propres actions. Mais comme le monde extérieur et les croyances des agents évoluent constamment, un agent doit aussi posséder un moyen pour déterminer si les engagements existants sont encore valides. Les conventions offrent un tel mécanisme en définissant les conditions sous lesquelles les engagements doivent être réévalués et en spécifiant les actions à entreprendre dans de telles circonstances.

Du point de vue du concepteur d'un SMA, diverses questions doivent être traitées comme notamment [FER 95] : avec quels agents un agent doit-il coordonner ses actions ? Quand et où ces actions de coordination doivent-elles être accomplies ? Comment détecter et traiter les interactions entre actions (conflits et renforcement) ? Détecter les relations existant entre des actions est une activité nécessaire lorsque l'on veut coordonner ces actions.

Von Martial [VMA 90] de son côté a identifié deux grandes catégories de relations pouvant exister entre les actions accomplies simultanément par plusieurs agents : les relations négatives et les relations positives. Les relations négatives (ou conflictuelles) sont celles qui gênent ou empêchent plusieurs actions de s'accomplir simultanément et sont dues en général à des incompatibilités de buts ou des conflits de ressources. Par exemple dans une vente aux enchères, agent X et agent Y

veulent acquérir un même meuble M . Les relations positives (ou synergiques) sont celles qui permettent aux actions de bénéficier les unes des autres. Ainsi, la réalisation d'une action a accomplie par l'agent Y réalise du même coup une action $x = b$ que devait accomplir l'agent X ou favorise la réalisation d'une action c par l'agent Z . Par exemple X , Y et Z sont dans une pièce dont les fenêtres sont fermées et les stores baissés. X a chaud et Z aimerait avoir de la lumière. Y monte les stores et ouvre une fenêtre.

Le lecteur intéressé par une étude détaillée de la coordination entre agents pourra se référer au chapitre 4. Dans ce chapitre, la coordination est vue plus particulièrement sous l'angle de la coordination d'actions par planification multi-agent.

1.4. Négociation entre agents

La négociation joue un rôle fondamental dans les activités de coopération en permettant aux personnes de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. Durfee et ses collègues [DUR 89] définissent la négociation comme le processus d'améliorer les accords (en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes. En général, les chercheurs en IAD utilisent la négociation comme un mécanisme pour coordonner un groupe d'agents. Différentes approches ont été développées en s'appuyant sur la riche diversité des négociations humaines dans divers contextes [CON 91b, DAV 83, ROS 94, SYC 89].

Un des protocoles les plus étudiés pour la négociation s'appuie sur une métaphore organisationnelle [DAV 83]. Le protocole du réseau contractuel (*Contract-Net*) a été une des approches les plus utilisées pour les SMA [SMI 80b]. Les agents coordonnent leurs activités grâce à l'établissement de contrats afin d'atteindre des buts spécifiques. Un agent, agissant comme un gestionnaire (*manager*) décompose son contrat (une tâche ou un problème) en sous-contrats qui pourront être traités par des agents contractants potentiels. Le gestionnaire annonce chaque sous-contrat sur un réseau d'agents. Les agents reçoivent et évaluent l'annonce. Les agents qui ont les ressources appropriées, l'expertise ou l'information requise envoient au gestionnaire des soumissions (*bids*) qui indiquent leurs capacités à réaliser la tâche annoncée. Le gestionnaire évalue les soumissions et accorde les tâches aux agents les mieux appropriés. Ces agents sont appelés des contractants (*contractors*). Enfin, gestionnaires et contractants échangent les informations nécessaires durant l'accomplissement des tâches. Par exemple, Parunak [VPA 96] a utilisé ce protocole pour développer un système de contrôle de production.

Un autre important protocole de négociation a été proposé par Cammarata et ses collègues [CAM 83, CAM 88]. Ils ont étudié les stratégies de coopération pour résoudre des conflits entre des plans d'un ensemble d'agents. Ces stratégies ont été appliquées au domaine du contrôle de trafic aérien pour permettre à chaque agent (représentant un avion) de construire un plan de vol qui permettrait de garder une distance sécuritaire par rapport aux autres avions et de satisfaire des contraintes telles que « atteindre la destination désirée avec une consommation de carburant minimale ». La stratégie choisie, appelée *centralisation de tâche* permettait aux agents impliqués dans une situation conflictuelle potentielle (des avions se rapprochant trop compte tenu de leurs caps respectifs) de choisir l'un d'eux pour résoudre le conflit. Cet agent agissait comme un planificateur centralisé et développait un plan multi-agent qui spécifiait les actions concurrentes de tous les avions impliqués. Les agents utilisaient la négociation pour déterminer qui était le plus apte à réaliser le plan. Cette aptitude était évaluée à partir de divers critères permettant d'identifier par exemple l'agent le mieux informé ou celui qui était le plus contraint. Les protocoles de négociation précédents supposent que les agents sont coopératifs, et donc qu'ils poursuivent un but commun.

Sycara [SYC 89] a étudié des situations dans lesquelles on ne peut pas supposer la coopération des agents. La résolution de conflits est mise en œuvre soit par négociation soit à l'aide d'un médiateur. Sycara a développé un système qui résout des conflits dans le domaine des relations de travail. Ce système comprenait trois agents : l'employeur, les employés et le médiateur. L'employeur et les employés avaient des buts conflictuels et le médiateur générait des propositions et contre-propositions pour essayer de rapprocher les deux parties adverses.

Müller de son côté [MUL 96] a indiqué que les recherches en négociation peuvent être divisées en trois catégories. Les recherches sur les langages de négociation qui s'intéressent aux primitives de communication pour la négociation, à leur sémantique et à leur usage dans les protocoles. Les recherches sur les décisions en négociation qui s'intéressent aux algorithmes pour comparer les sujets de négociation, les fonctions d'utilité et la caractérisation des préférences des agents. Les recherches sur le processus de négociation qui étudient des modèles généraux des comportements de négociation des agents.

1.5. Planification dans un environnement multi-agent

Bond et Gasser [BON 88] font remarquer que nous pouvons obtenir une plus grande coordination si l'on oriente les comportements des agents vers des buts communs en établissant explicitement une division du travail entre les agents. Des techniques comme la planification centralisée pour des groupes d'agents, la conciliation de plans, la planification distribuée, l'analyse organisationnelle, sont des

façons d'aider les agents à aligner leurs activités en assignant les tâches après avoir raisonné sur les conséquences de réaliser ces tâches dans des ordres particuliers. Dans une approche de planification multi-agent, un plan multi-agent est un plan qui est créé afin que plusieurs agents puissent l'exécuter. La création d'un tel plan peut être faite par un seul ou par plusieurs agents. En planification multi-agent centralisée, un agent est responsable pour la création du plan qui spécifie les actions planifiées pour tous les agents concernés. C'est cette approche qui a été utilisée par Cammarata pour le problème de contrôle de trafic aérien.

Une autre façon d'implémenter la planification multi-agent centralisée a été proposée par Georgeff [GEO 83]. Dans cette approche, les plans des agents sont d'abord créés de façon individuelle. Ensuite un agent centralisateur rassemble ces plans et les analyse pour identifier les conflits. L'agent centralisateur en question tente de résoudre les conflits en modifiant les plans locaux des autres agents et en introduisant des commandes de communication afin que les agents se synchronisent de façon appropriée. Dans une approche de planification distribuée, les activités de planification sont réparties au sein d'un groupe d'agents. Cette approche est utilisée quand un seul agent ne peut pas avoir une vue globale des activités du groupe.

Von Martial [VMA 90] a distingué deux classes de problèmes en planification distribuée : la planification dirigée par les tâches et la coordination de plans. Dans les systèmes de planification dirigés par les tâches, il y a un but initial ou une tâche initiale qui est décomposée en sous-buts ou sous-tâches et qui est répartie entre plusieurs agents (décomposition de problème *top-down*). Par ailleurs, la coordination de plan s'attaque à des situations dans lesquelles il existe des plans d'agents (vision centrée sur les agents) et le problème consiste à réconcilier ces plans avant de les exécuter dans un environnement commun. En général, la planification multi-agent nécessite une forme ou une autre de synchronisation de plans qui peut être réalisée à divers moments : pendant la décomposition de plan, pendant la construction de plan ou après celle-ci. Les plans des agents peuvent être en conflits en raison d'incompatibilités d'états des systèmes, de l'ordre des activités ou de l'usage des ressources. De tels conflits peuvent être résolus par un agent en particulier (coordonnateur ou médiateur) ou une solution peut être obtenue par négociation.

Par la suite, Von Martial [VMA 92] a proposé un modèle de coordination de plans qui s'appuie sur la résolution de conflits basée sur les divers types de relations (négatives et positives) pouvant exister entre ces plans d'une part, et sur un protocole de communication pour la synchronisation des plans d'autre part. Les relations entre plans entraînent une modification des plans des agents. Les relations négatives sont traitées pour résoudre les conflits et les relations positives sont utilisées pour tirer parti de plans convergents.

Conry et ses collègues [CON 89, CON 91a] ont développé un protocole de négociation appelé *multi-stage negotiation* pour résoudre de façon coopérative des conflits dans l'allocation de ressources. Le domaine d'application était celui de la surveillance et du contrôle dans un système de communication complexe. Le protocole débute par la génération d'un plan initial et consiste en plusieurs cycles permettant d'envoyer des requêtes pour des buts secondaires, l'analyse locale, la génération de plans alternatifs et l'envoi de réponses.

Durfee et Lesser [DUR 87, DUR 89] ont proposé une approche appelée *planification partielle globale* (PGP pour *Partial Global Planning*) dans laquelle les agents construisent des plans et les partagent pour identifier des améliorations potentielles pour leur coordination. Au contraire de la planification multi-agent qui suppose qu'un plan est élaboré avant que les agents commencent à agir, la planification partielle globale permet aux agents d'alterner phases de planification et d'exécution : les agents commencent par planifier des interactions coordonnées au mieux de leurs connaissances et réagissent aux situations non prévues en modifiant leurs plans. Cette approche a été appliquée à une plate-forme de tests multi-agents appelée DVMT (*Distributed Vehicle Monitoring Testbed*). Dans ce contexte, Lesser [LES 91b] a utilisé le modèle FA/C (*Functionally Accurate/Cooperative mode*). Il a développé ce modèle pour répondre aux déficiences des modèles conventionnels qui supposent que les tâches sont décomposées de façon à ce que chaque agent ait suffisamment de données pour résoudre complètement et précisément les sous-problèmes qui lui sont assignés, et ce avec peu d'interaction avec les autres agents. Toutefois, dans de nombreuses applications (interprétation distribuée, planification distribuée et allocation de ressources, etc.) ce modèle de décomposition de tâches est inadéquat. Dans le modèle FA/C, les agents n'ont pas besoin de disposer de toutes les informations localement pour résoudre leurs sous-problèmes. Ils interagissent en échangeant des résultats partiels. Dans un SMA, les agents doivent gérer des ressources distribuées qui peuvent être physiques (capacités de communication, matières premières, argent) ou informationnelles (telles que les informations relatives à la décomposition du problème). Les agents doivent adapter leurs plans pour tenir compte de la disponibilité des ressources.

Bien entendu, la planification contribue à la coordination, dans la mesure où lorsque les agents adoptent un plan « bien fait », ils agissent généralement de manière coordonnée. A. El-Fallah Seghrouchni [ELF 95, ELF 96] a travaillé sur ce type d'approche dont les éléments essentiels sont repris au chapitre 4.

1.6. Communication entre agents

Les agents peuvent interagir, soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non linguistiques qui

modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités. Dans les SMA, deux stratégies principales ont été utilisées pour supporter la communication entre agents : les agents peuvent échanger des messages directement ou accéder à une base de données partagées (appelée tableau noir ou *blackboard*) dans laquelle les informations sont postées. Les communications sont à la base des interactions et de l'organisation sociale d'un SMA.

Dans certains cas, un agent peut inférer les plans des autres agents sans avoir à communiquer avec eux. Dans [TUB 84] ce mécanisme est appelé *marchandage tacite* et il semble fonctionner au mieux quand les buts des agents ne sont pas conflictuels. Pour étudier ce mécanisme, Rosenschein et ses collègues [ROS 86] ont utilisé une approche de la théorie des jeux caractérisée par des matrices de gains qui contiennent les gains des agents pour chacune des issues possibles de l'interaction. Cette approche suppose que les agents partagent la connaissance de cette matrice de gains, ce qui n'est pas réaliste si les agents ne sont pas bienveillants. Rosenschein [ROS 85a] a étendu cette approche afin de permettre aux agents de raffiner leurs choix en tenant compte de l'incertitude des choix des autres agents. Cette approche présente aussi des difficultés car le fait que les agents aient à spéculer sur les choix des autres agents conduit à développer des mécanismes de raisonnement complexes qui peuvent nécessiter une quantité de calculs importante. Il est clair que l'absence de communication doit être compensée par des mécanismes de raisonnement plus complexes. Certains chercheurs ont examiné l'utilisation de communications primitives restreintes à l'utilisation d'un ensemble fini de signaux (souvent au nombre de deux) avec des interprétations fixes comme celles de Hoare [HOA 78]. Cette stratégie a été employée par Georgeff [GEO 83] pour éviter des conflits dans une planification multi-agent. Cependant, des informations plus sophistiquées comme des requêtes ou des commandes ne peuvent pas être exprimées avec de tels signaux.

1.6.1. Transfert de plans ou de messages

Dans l'approche de transfert de plans, un agent *X* communique son plan en totalité à un agent *Y* et *Y* communique son plan en totalité à *X*. Cette approche présente plusieurs inconvénients : le transfert de plans est coûteux en ressources de communication et c'est une approche difficile à mettre en œuvre dans des applications réelles car il y a en général un fort degré d'incertitude au sujet des états actuels et futurs du monde. De plus, dans des situations réelles, il n'est pas possible de formuler à l'avance des plans en totalité. Par conséquent, on a besoin de communiquer des stratégies générales aux agents plutôt que des plans, et donc il est nécessaire que les agents puissent échanger des messages. De très nombreux travaux ont mis en œuvre le transfert de message associé à l'usage de protocoles précis. Dans un SMA la complexité des communications dépend des caractéristiques des

agents. Des agents réactifs utilisent un ensemble de règles de communication qui sont déclenchées quand des états spécifiques sont vérifiés ou quand des événements prédéterminés se produisent. Ces événements peuvent être des messages reçus ou des changements dans l'environnement. Les types de messages combinés avec les réponses attendues conduisent à la spécification de protocoles d'interaction. Comme on l'a vu, de tels protocoles ont été développés pour divers types de systèmes comme le protocole du réseau contractuel [SMI 80a] ou des protocoles de coopération [CAM 83]. Les communications et l'exécution d'actions sont en général intercalées. Aussi des agents intentionnels doivent être capables de construire des plans dans lesquels des actions linguistiques et non linguistiques sont combinées. Kreifelts et Von Martial [KRE 91] ont proposé un protocole de négociation de ce type.

1.6.2. *Echange d'informations grâce à un tableau noir*

En intelligence artificielle la technique du tableau noir (*blackboard*) est très utilisée pour spécifier une mémoire partagée par divers systèmes [NII 89a, NII 89b]. Dans un SMA utilisant un tableau noir, les agents peuvent écrire des messages, insérer des résultats partiels de leurs calculs et obtenir de l'information. Le tableau noir est en général partitionné en plusieurs niveaux qui sont spécifiques à l'application. Les agents qui travaillent sur un niveau particulier peuvent accéder aux informations contenues dans le niveau correspondant du tableau noir ainsi que dans des niveaux adjacents. Ainsi, les données peuvent être synthétisées à n'importe quel niveau et transférées aux niveaux supérieurs alors que les buts de haut niveau peuvent être filtrés et passés aux niveaux inférieurs pour diriger les agents qui oeuvrent à ces niveaux. Le transfert de messages et les communications basées sur un tableau noir sont souvent combinés dans des systèmes complexes. Dans de tels systèmes, chaque agent est composé de plusieurs sous-systèmes (ou *sous-agents*) qui communiquent à travers un tableau noir. Les agents communiquent entre eux par échange de messages. Plusieurs MAS ont été bâtis sur ce genre d'architecture comme le DVMT (*Distributed Vehicle Monitoring Testbed*) [LES 80] ou le système MINDS [HUH 87]. Dans le système MINDS, chaque usager travaille avec un agent qui utilise un tableau noir pour retrouver des documents dans sa base de données locales ou communique par messages avec d'autres agents du même type.

1.6.3. *Actes de discours et conversations*

Dès lors que l'on s'intéresse au contenu des messages échangés par les agents, il est pertinent d'examiner les nombreux travaux qui ont été réalisés sur la compréhension du langage naturel et en particulier les recherches sur l'intentionnalité dans les communications [COH 90a]. Dans ces travaux, on considère que les agents sont dotés de structures de données appelées *états mentaux*

à partir desquelles ils peuvent raisonner. Diverses catégories d'états mentaux ont été retenues par les chercheurs dont notamment les croyances, les désirs, les intentions ou buts, les capacités, etc. Ce sont ces mêmes états mentaux qui caractérisent les modèles des agents cognitifs en intelligence artificielle distribuée. Ainsi, un agent cognitif raisonne sur ses croyances qui représentent sa compréhension du monde dans lequel il évolue. Il raisonne aussi sur ses désirs et intentions en relation avec ses croyances et capacités afin de prendre des décisions auxquelles sont associés les plans qu'il va accomplir pour agir dans le monde. Lorsqu'un agent *X* communique avec un autre agent *Y*, c'est pour influencer les états mentaux de *Y*. Ainsi, *X* peut transférer à *Y* une information qui va changer les croyances de *Y* ou bien lui proposer d'adopter un but que *Y* va transformer en une de ses intentions. Il est donc pertinent de s'intéresser à la façon dont les messages échangés peuvent être structurés pour refléter ces mécanismes d'influence sur les états mentaux des agents.

Les philosophes du langage ont développé au cours des 25 dernières années la *théorie des actes de discours* qui considère que « dire quelque chose c'est en quelque sorte agir » [SEA 69]. Les paroles sont interprétées comme des actions appelées *actes de discours*. Cette théorie fournit un cadre qui identifie des types d'actes de discours primitifs et permet de dériver de ces types primitifs l'interprétation sous une forme logique de n'importe quel verbe du langage exprimant un acte de discours quelconque. Les chercheurs en intelligence artificielle ont très tôt pensé à étudier comment les actes de discours peuvent être utilisés pour influencer les états mentaux des agents. Cohen et Perrault ont travaillé sur la reconnaissance et la génération d'actes de discours en utilisant des techniques de planification [COH 79].

Allen [ALL 80] suggéra qu'un locuteur peut identifier des actes de discours directs et indirects s'il est capable de reconnaître le plan de son interlocuteur ainsi que les obstacles qui empêchent sa réalisation. Cependant, cette approche nécessite que l'agent soit capable d'identifier avec précision les catégories d'actes de discours correspondant aux énonciations de ses interlocuteurs, ce qui n'est pas toujours facile comme le remarquent Cohen et Levesque [COH 90b]. Cohen et Levesque [COH 79, COH 90a] ont proposé une théorie permettant de tenir compte des états mentaux dans le processus de communication. Ils partent d'une théorie des interactions rationnelles basée sur les intentions et engagements des agents. Cette approche, qui a influencé grandement les recherches dans ce domaine, suppose que les agents sont bienveillants. Galliers [GAL 88] a étendu cette théorie à des situations mettant en jeu des agents qui peuvent être en conflit, en montrant qu'il n'est pas nécessaire de considérer que les agents sont bienveillants. Parallèlement, Werner [WER 89] a proposé une théorie unifiant des techniques de communication, de coopération et de structuration sociale pour la conception de systèmes d'agents qui adoptent des comportements de groupe : une formalisation des états intentionnels des agents en relation avec les actes de discours et leurs effets sur le processus de planification.

Shoham [SHO 92] a proposé une approche formelle décrivant diverses lois sociales applicables aux sociétés d'agents. Castelfranchi et Conte [CAS 96] soulèvent diverses objections aux fondements sociologiques actuels de l'IAD : une emphase trop grande est mise sur la coopération tant au niveau des buts partagés, des actions conjointes et du partage d'information ; les agents sont *hypercognitifs* dans le sens que l'on dote les agents de capacités d'omniscience, d'introspection qui ne sont pas socialement plausibles ; on néglige les mécanismes équivalents à des lois, règlements ou normes qui limitent la liberté des agents ; on néglige aussi le rôle de la communication entre les agents comme moyen d'influencer les autres. Ces auteurs proposent une voie alternative dans laquelle on mettrait l'emphase sur les dépendances sociales existant entre agents et sur leur influence dans la formation de buts communs. Pour adopter un comportement social, un agent doit être capable non seulement de raisonner sur ses états mentaux mais aussi de prendre en compte les connaissances qu'il partage avec les autres agents (*mutual belief*) et les engagements qui les relient [COH 90b, RAO 91b, SIN 94]. On mesure ainsi l'importance d'associer à un modèle de la communication interagent un modèle de raisonnement sur les états mentaux, tout en tenant compte de la dimension sociale de l'interaction. Les relations sociales ont aussi un impact évident dans les conversations humaines : la forme et le contenu des actes de discours des locuteurs tiennent compte des rôles et statuts sociaux des participants à la conversation.

Cet aspect n'a été exploré que récemment dans le cadre de la modélisation des interactions et la simulation des conversations entre agents logiciels [BOU 98]. Barbuceanu et Fox [BAR 95] considèrent une activité de coordination entre agents comme un genre de « conversation » qui est décrite par un graphe d'états finis : les états du graphe représentent les états que peut prendre la conversation. Un ensemble de règles de conversation spécifient comment un agent qui est dans un état conversationnel donné reçoit un message d'un type particulier (spécifié suivant les conventions de KQML), réalise des actions locales (par exemple : « Mettre à jour sa base de données locale »), envoie des messages et passe dans un autre état conversationnel.

De leur côté, Rousseau et Moulin [ROU 95] ont proposé une approche de simulation des conversations entre agents vue comme un processus de négociation multiniveau au cours duquel les participants négocient au sujet de leurs états mentaux (croyances, désirs, intentions, émotions, etc.). Dans cette approche les actes de discours sont interprétés par les agents en termes d'objets de conversation (OC) qui sont des états mentaux associés à des positionnements d'agents (proposition, acceptation, refus). Les OC proposés par les agents locuteurs au cours de la conversation forment un réseau de concepts qui constitue une mémoire persistante de la conversation partagée par les agents locuteurs et à partir de laquelle ils peuvent raisonner.

1.6.4. Communication utilisant KQML, ACL et les conversations

Le langage KQML [FIN 94] a été proposé pour supporter la communication interagent. Ce langage définit un ensemble de types de messages (appelés abusivement *performatifs*) et des règles qui définissent les comportements suggérés pour les agents qui reçoivent ces messages. Les types de messages de KQML sont de natures diverses : simples requêtes et assertions (par exemple ; *ask*, *tell*) ; instructions de routage de l'information (*forward* et *broadcast*) ; commandes persistantes (*subscribe*, *monitor*) ; commandes qui permettent aux agents consommateurs de demander à des agents intermédiaires de trouver les agents fournisseurs pertinents (*advertise*, *recommend*, *recruit* et *broker*). Comme le font remarquer Cohen et Levesque [COH 95], ce langage a été développé de façon *ad hoc* pour les besoins des développeurs d'agents logiciels : le terme « performatif » a été utilisé pour nommer diverses commandes qui ont une certaine ressemblance avec des verbes utilisés de façon performative dans le langage naturel. L'interprétation sémantique actuelle de ces commandes n'est pas satisfaisante.

Ces dernières années, KQML semble perdre du terrain au profit d'un autre langage plus riche sémantiquement : ACL (pour *Agent Communication Language*). Un langage mis en avant par la FIPA, qui s'occupe de standardiser les communications entre agents. ACL est basé également sur la théorie du langage et a bénéficié grandement des résultats de recherche de KQML. Si toutefois les deux langages se rapprochent au niveau des actes du langage, il n'en est rien au niveau de la sémantique et il semble qu'un grand soin a été apporté au niveau de ACL tant au niveau de certains protocoles, qui sont plus explicites, qu'au niveau de la sémantique des actes eux-mêmes.

Nous pensons qu'il serait plus approprié de réviser le langage KQML et ACL en utilisant une sémantique faisant appel à la recursion des actes du langage, telle que proposée par Chaib-draa et Vanderveken [CHA 99], et un cadre théorique d'interprétation des actes de discours tel que proposé par Vongkasem [VON 99].

Nous n'avons fait qu'effleurer les communications et le lecteur intéressé par plus de détails, en particulier en ce qui concerne KQML et ACL, pourra se référer au chapitre 3.

1.7. Méthodes de conception des systèmes multi-agents

Les SMA sont souvent des systèmes complexes qui demandent de longs efforts de développement. Du début des années 1980 jusqu'au milieu des années 1990, les chercheurs se sont surtout concentrés sur l'exploration de nouveaux concepts, la mise en place de théories, la réalisation de divers prototypes de SMA. Ce

bouillonnement d'idées est tout à fait normal dans un jeune domaine technologique : c'est ce que l'on a observé pour les bases de données dans les années 1970, les systèmes experts et les systèmes orientés objets dans les années 1980. Cependant, lorsque la technologie atteint plus de maturité et de stabilité, il devient nécessaire de développer des méthodes de conception qui permettent aux entreprises de développer de façon systématique des systèmes opérationnels dans un environnement industriel. Telle est l'évolution que l'on observe pour les SMA depuis quelques années. Parmi les travaux des pionniers, citons Decker et ses collègues [DEC 89], qui ont proposé un cadre de référence pour évaluer les recherches en résolution coopérative de problèmes distribués (*cooperative distributed problem solving*). Leur but était de comprendre les relations entre des efforts de recherche disparates et d'évaluer les progrès dans ce champ de recherche. Bien que ce cadre ne soit pas une méthode de conception, il soulignait un besoin de prendre du recul par rapport au foisonnement des recherches.

Werner [WER 89] proposa lui aussi un cadre de référence pour essayer de comprendre les « relations qui peuvent s'établir entre les usagers, les développeurs et le SMA en construction ». Cette réflexion essayait de situer le développement traditionnel de systèmes informatiques par rapport au développement d'un SMA dans lequel les agents pourraient évoluer par eux-mêmes, choisissant leur propres buts, créant leurs propres plans, etc. Ce cadre n'était pas une méthode de conception à proprement parler, mais dénotait une préoccupation méthodologique intéressante. Goodson et Schmidt [GOO 90] ont proposé la première démarche méthodologique visant à décomposer un problème multi-agent en des unités fonctionnelles, d'identifier les unités fonctionnelles que la machine puisse traiter mieux que les usagers et de construire les modules de résolution de problèmes pour ces unités. Moulin et Cloutier [MOU 94] ont développé, quant à eux, la méthode MASB (*Multi-Agent Scenario-Based method*), une méthode de conception de SMA basée sur l'analyse et la conception de scénarios caractérisant les rôles joués par les agents humains et les agents artificiels et qui s'applique bien pour des applications de *travail collaboratif* (par exemple, un système distribué de prise de rendez-vous).

Burmeister [BUR 96] a proposé une méthode dans laquelle on utilise trois modèles pour analyser un SMA : un modèle d'agent (les agents et leur structure interne définie en termes de croyances, buts, plans, etc.), un modèle organisationnel (description des relations entre agents) et un modèle de coopération (description des interactions entre agents). Cette méthode empruntait des techniques de modélisation aux méthodes orientées objet telles que les cartes CRC et la notation d'OMT. Kinny et ses collègues [KIN 96] ont proposé une méthode qui distingue deux niveaux de conception (interne et externe) pour la création d'agents BDI. Les agents BDI mettent en œuvre un modèle basé sur les notions de croyance (*Belief*), désir (*Desire*) et intention (*Intention*). Ce modèle a été popularisé par Rao et Georgeff [RAO 91a, RAO 91b, RAO 92]. Le niveau externe de ce modèle est spécifié par un modèle qui

décrit les relations hiérarchiques entre agents et un modèle d'interaction qui définit les responsabilités, services et interactions entre les agents et leur environnement. Le niveau interne permet de modéliser chaque agent BDI grâce à trois modèles : un modèle de croyances, un modèle de buts et un modèle de plans.

Kendall et ses collègues [KEN 96] ont proposé une approche de conception de SMA qui combine des techniques orientées objets et des techniques de modélisation d'entreprise et de fabrication assistée par ordinateur. Dans cette méthode, les auteurs ont essayé d'adapter les techniques de modélisation des systèmes orientés objets au domaine des systèmes multi-agents (*use cases*, diagrammes de scénarios, patrons de conception, etc.). De leur côté, Maamar et Moulin [MAA 97] ont proposé une méthode de conception de systèmes composés d'agents hétérogènes en proposant l'utilisation de cadres (*frameworks*) orientés agents. L'idée est de spécifier les diverses composantes de systèmes distribués et hétérogènes sous la forme d'équipes d'agents dans lesquelles chaque agent joue des rôles spécifiques. Les *frameworks*, les équipes et les agents sont caractérisés par les services qu'ils rendent et ceux qu'ils requièrent. Diverses techniques de modélisation sont employées pour spécifier les scénarios, les diagrammes d'interaction des agents, les modèles d'agents. Cette méthode a été utilisée pour la réalisation d'un prototype permettant de réaliser l'interopérabilité de géorépertoires.

La méthode Cassiopea [COL 96] distingue trois phases pour la conception d'un SMA et a été appliquée au domaine des « robots footballeurs ». Cette méthode utilise des techniques orientées objets pour spécifier les comportements élémentaires des agents. Les comportements relationnels entre agents sont par la suite analysés grâce à un graphe de couplages. Enfin, on spécifie la dynamique de l'organisation des agents (qui s'allie avec qui ?) en utilisant le graphe de couplages.

Verharen [VER 97] a proposé une méthode de conception qui s'appuie sur une perspective de gestion des activités d'un SMA et met notamment en œuvre les modèles suivants : un modèle d'autorisation qui décrit les types de communications permises et les obligations qui existent entre une organisation et son environnement ; un modèle de communication qui raffine le modèle précédent sous la forme de contrats entre les agents sous la forme de réseaux de Petri ; les transactions entre les agents sont modélisées par des diagrammes de transactions qui associent actes de discours et buts des agents ; un modèle de tâches qui fournit la décomposition des tâches à partir des diagrammes précédents.

Les méthodes CoMoMAS [GLA 96] et MAS-CommonKADS [IGL 97] proposent des extensions à la méthode CommonKADS [VER 97], qui a été développée pour le domaine de l'ingénierie des connaissances, en s'appuyant sur certaines techniques orientées objets (*use cases*, scénarios, modèles de classes). Ces méthodes définissent divers modèles : le modèle d'agent qui décrit les caractéristiques des agents (capacités

de raisonnement, buts, services); le modèle de tâches (tâches et sous-tâches); le modèle d'expertise (connaissances nécessaires aux agents pour réaliser leurs tâches); le modèle de coordination (description des interactions possibles entre agents, protocoles et capacités nécessaires); le modèle d'organisation qui décrit comment le SMA va être introduit dans l'organisation qui accueille la société d'agents; le modèle de communication qui décrit les interactions entre les usagers et les agents; le modèle de conception qui synthétise les modèles précédents en tenant compte des contraintes d'implantation. Ces méthodes ont été utilisées dans divers projets de recherche.

Comme on peut le constater, plusieurs méthodes d'analyse et de conception des SMA ont été proposées récemment. La plupart d'entre elles s'appuie sur des techniques de modélisation empruntées à des méthodes connues en développement orienté objets ou en ingénierie des connaissances pour aider à la construction des modèles d'agents, de l'architecture du SMA, pour la spécification des modèles d'organisation, d'interaction, etc. Peut-on s'attendre à la création d'une approche standard comme cela fut le cas en ingénierie des connaissances avec la méthode KADS ou à une normalisation des modèles comme ce fut le cas en orienté objets avec le langage UML ? L'avenir nous le dira. Une telle standardisation apparaîtra certainement quand les technologies multi-agents seront adoptées effectivement par les entreprises.

Précisons, pour finir, que Guessoum et Occello dans le chapitre 5 reviennent en détail sur les environnements de développement des systèmes multi-agents.

1.8. Quelques exemples d'application des SMA

De nos jours, la technologie multi-agent a trouvé sa place dans les systèmes manufacturiers, les systèmes financiers, les loisirs, les télécommunications, le contrôle-commande, les systèmes embarqués, et beaucoup d'autres applications.

Dans ce qui va suivre, nous n'exposerons que quelques applications utilisant cette technologie et nous renvoyons le lecteur à l'article de Jennings [JEN 98a] et aux chapitres 6 et 7 du présent volume pour un ensemble d'exemples plus complet.

Plus précisément, nous présentons ici un domaine d'application très riche pour les agents : les télécommunications et trois autres systèmes, l'un œuvrant dans le domaine de la gestion du processus d'affaires, le système ADEPT, le deuxième dans le domaine médical, le système GUARDIAN, et finalement le dernier dans les environnements riches en information, le système NETSA.

1.8.1. Application des SMA aux télécommunications

Ces dernières années, les télécommunications ont notamment introduit une conception de services décentralisée dans le contexte du *web*, créé de nouveaux services de médiation tels que les portails et engendré l'apparition de nombreux fournisseurs de services réseaux qui ne disposent pas de leurs propres services réseaux [BOU 01a]. L'obtention de tels services décentralisés ne peut, bien entendu, être obtenue que grâce à des logiciels pour lesquels les données et le contrôle sont forcément distribués. De ce fait, il est clair que les SMA semblent convenir aux télécommunications. C'est pourquoi les principaux acteurs des télécommunications mènent actuellement d'intenses activités de recherche sur la technologie agent : British Telecom, France Télécom, Deutch Telekom, NTT, Nortel, Siemens, etc. (voir chapitre 10).

1.8.2. Le système ADEPT

Les gestionnaires de grandes compagnies prennent des décisions en se basant sur une combinaison de jugements et d'informations provenant de plusieurs départements. Idéalement, toutes les informations pertinentes devraient être rassemblées avant qu'une décision ne soit prise. Cependant, le processus pour obtenir des informations à jour et pertinentes est très complexe et prend énormément de temps. Pour cette raison, plusieurs compagnies ont cherché à développer des systèmes informatiques afin de les assister dans leur processus d'affaires.

Le système ADEPT [ALT 94] traite ce problème en voyant le processus d'affaires comme un ensemble d'agents qui négocient et qui offrent des services. Chaque agent représente un rôle distinct ou un département de l'entreprise et est en mesure de fournir un ou plusieurs services. Les agents qui requièrent les services d'autres agents le font par une négociation qui permet d'obtenir un coût, un délai temporel et un degré de qualité acceptables par les deux parties. Le résultat d'une négociation terminée avec succès constitue un engagement entre les deux parties.

1.8.3. Le système GUARDIAN

Le système GUARDIAN [ROT 89] a pour but de gérer les soins aux patients d'une unité chirurgicale de soins intensifs. Les principales motivations de ce système sont, premièrement, que le modèle des soins d'un patient dans une unité de soins intensifs est essentiellement celui d'une équipe, où un ensemble d'experts dans des domaines distincts coopèrent pour organiser les soins des patients. Deuxièmement, le facteur le plus important pour donner de bons soins aux patients est le partage d'informations entre les membres de l'équipe. Particulièrement, les

médecins spécialistes n'ont pas l'opportunité de superviser l'état d'un patient minute par minute, cette tâche revient aux infirmières qui, souvent, ne possèdent pas les connaissances nécessaires à l'interprétation des données qu'elles rassemblent.

Le système GUARDIAN répartit donc le suivi des patients entre un certain nombre d'agents de trois types différents. Les agents *perception/action* sont responsables de l'interface entre GUARDIAN et le monde environnant, établissant la relation entre les données des senseurs et une représentation symbolique que le système pourra utiliser, et traduisant les requêtes d'action du système en commandes pour les effecteurs³. Les agents en charge du raisonnement sont responsables d'organiser le processus de prise de décision du système. Finalement, les agents en charge du contrôle (il n'y en a habituellement qu'un seul) assurent le contrôle de haut niveau du système.

1.8.4. Les systèmes d'information coopératifs (SIC)

Les SIC sont généralement caractérisés par la grande variété et le grand nombre de sources d'information. Ces sources d'information sont hétérogènes et distribuées soit sur un réseau local (intranet) soit sur l'internet. De tels systèmes doivent être capables d'exécuter principalement les tâches suivantes :

- la découverte des sources : trouver la bonne source de données pour l'interroger ;
- la recherche d'informations : identifier les informations non structurées et semi-structurées ;
- le filtrage des informations : analyser les données et éliminer celles qui sont inutiles ;
- la fusion des informations : regrouper les informations d'une manière significative.

Le système multi-agent Warren pourrait constituer un exemple spécifique de l'utilisation des agents dans ce type d'application. C'est un système d'agents intelligents pour l'aide à la gestion des portefeuilles [ZEN 96]. Ce système combine les données du marché financier, les rapports financiers, les modèles techniques et les rapports analytiques avec les prix courants des actions des compagnies. Toutes ces informations sont déjà disponibles sur le *web*, Warren ne fait que les intégrer *via* des agents spécialisés, les agents d'information et ensuite les présenter aux usagers. Pour ce faire, Warren dispose de six agents ressources, deux agents de tâches et un agent utilisateur pour chaque usager. L'agent utilisateur affiche (*via* le *web*) les

3. Traduction du terme anglais *effectors*.

informations financières de son usager lui permettant de faire des simulations d'achat et de vente des actions. Il affiche également les prix courants des actions et les nouvelles informations du marché financier. Le même agent permet également d'accéder aux rapports produits par les deux agents de tâches. Ces deux agents fournissent, d'une part, une intégration graphique des prix et des nouvelles concernant les actions et, d'autre part, une analyse fondamentale des actions en tenant compte de leurs historiques. Les agents d'information accèdent à différentes sources d'information, comme les pages *web*, les nouvelles de Clarinet et Dow-Jones, les rapports financiers électroniques de SEC Edgar ainsi que d'autres rapports sous un format texte. Warren n'est qu'un exemple parmi d'autres et il existe actuellement plusieurs autres systèmes qui traitent ce genre d'application. Parmi ces applications, nous pouvons citer :

- Infosleuth : c'est un système multi-agent pour la recherche coopérative d'informations dans des bases de données distribuées. Ce système a été appliqué au domaine médical [NOD 98] ;
- NetSA (pour *Networked Software Agents*) : c'est un système proche de Infosleuth et dédié aux environnements riches en informations [COT 98, COT 99] ;
- UMDL : c'est un système d'information coopératif pour la recherche des documents dans une librairie digitale [VID 98].

A titre d'exemple, le système NETSA est un système multi-agent coopératif, développé à l'université Laval [COT 98, COT 99] et destiné aux environnements riches en informations. Ce système comporte plusieurs types d'agents :

- un agent *utilisateur* en charge de la cueillette et du filtrage des informations provenant et allant vers l'utilisateur ;
- un agent *courtier* servant de répertoire pour les agents qui évoluent au sein de NETSA ;
- des agents *ressources* reliés chacun à une ressource d'information et pouvant rapatrier et mettre à jour les données ;
- un agent *d'exécution* en charge de la décomposition des tâches et du suivi du déroulement d'exécution des différentes sous-tâches ;
- un agent *ontologie* en charge du maintien de la cohérence des concepts utilisés par les agents.

Les agents et les systèmes multi-agents sont utilisés dans plusieurs applications, comme par exemple : la gestion des réseaux, la recherche d'informations, le commerce électronique et la planification des tâches. Pour une grande partie de ces applications, les agents sont utilisés dans les systèmes d'information coopératifs ou comme assistants personnels dans certains travaux. Ce type d'applications nécessite des recherches approfondies et il convient en particulier :

- de pousser l'ingénierie de la construction des systèmes à agents collaboratifs. Il faut concevoir plus de méthodes et d'outils pour faciliter leurs implantations [SMI 94] ;
- de maîtriser et de bien identifier la coordination entre les agents. Il s'agit en particulier d'établir une théorie claire et formelle pour cette coordination [ROS 94] ;
- d'assurer les critères de stabilité, de clarté et de performance pour de tels systèmes ;
- de trouver des techniques qui permettent l'évaluation, la vérification et la validation de ces systèmes.

1.9. Conclusion

Nous avons vu tout au long de ce chapitre que la technologie agent et multi-agent n'est pas un concept voué à rester sur les tablettes des laboratoires de recherche puisque plusieurs applications existent déjà.

Les personnes qui ont développé des SMA diront toutefois qu'il est difficile de concevoir et de bâtir un système multi-agent. En effet, la construction de ce type de système comporte toutes les difficultés inhérentes aux systèmes répartis, auxquelles s'ajoute le caractère flexible et sophistiqué des interactions entre agents. De plus, les concepteurs de SMA font de nos jours face à deux difficultés majeures. Tout d'abord, l'absence de méthodologie systématique qui permettrait de spécifier et de structurer une application multi-agent. Ensuite, le manque d'outils commerciaux pour bâtir des SMA.

Une des forces importantes derrière la croissance rapide que connaissent aujourd'hui les systèmes multi-agents est l'internet, où la population d'agents est sans cesse croissante. Ces agents devront savoir collaborer afin d'atteindre les objectifs de leurs concepteurs. Dans ce type d'environnement, les agents rencontrent deux défis majeurs : ils doivent être en mesure de se rencontrer et d'interopérer. Une première solution à ce problème est l'introduction d'agents intermédiaires (*brokers*, *facilitateurs*, etc.) [COT 98, COT 99, TRO 99]. Ces agents intermédiaires ont pour principales fonctions : d'associer au mieux les besoins des utilisateurs et les services des fournisseurs ; d'unifier et probablement traiter les réponses des fournisseurs pour produire un résultat approprié ; d'aviser régulièrement les utilisateurs du changement des informations.

Dans cette optique d'interopérabilité, un grand nombre de langages de communication pour les agents ont été développés ; basés, pour plupart, sur la théorie des actes du discours. Bien que les performatifs offerts par ces langages permettent de caractériser les types des messages, ils ne permettent pas encore aux

agents de comprendre explicitement les concepts « discutés ». Le problème des ontologies reste donc ouvert.

Un autre problème critique est l'allocation de ressources limitées à un grand nombre d'agents. Des mécanismes basés sur les principes économiques ont été proposés pour résoudre ce problème. Dans de telles approches, on suppose que les agents sont égocentrés et ne cherchent qu'à maximiser leur utilité. Les sous-domaines où l'on a appliqué ces principes économiques sont l'allocation de ressources, l'allocation de tâches et la négociation. Là aussi la recherche n'a fait qu'entrevoir des techniques rudimentaires basées sur les lois du marché et il reste bien des domaines à couvrir comme par exemple les ventes aux enchères, les comportements acheteur(s) vendeur(s), le partage du marché, etc.

Finalement, il convient de préciser que les chercheurs travaillant sur les SMA d'un point de vue formel ont presque toujours fait appel à des agents omniscients, c'est-à-dire des agents ayant des capacités de raisonnement illimitées. Là aussi, le problème est ouvert et il y a aujourd'hui beaucoup d'équipes qui y travaillent.

Comme on a pu le constater, le domaine des systèmes multi-agents demeure encore aujourd'hui un domaine rempli de défis à surmonter, autrement dit un domaine très ouvert pour la recherche.

1.10. Bibliographie

- [AGH 86] AGHA G., *Actors: A Model Of Concurrent Computation In Distributed Systems*, MIT Press, Cambridge, Massachusetts, 1986.
- [AGH 88] AGHA G., HEWITT C., « Concurrent programming using actors », dans A. Yonezawa et M. Tokoro (dir.), *Object-Oriented Concurrent Programming*, p. 37-53, MIT Press, Computer Systems Series, 1988.
- [ALL 80] ALLEN J.F., PERRAULT C.R., « Analyzing intention in dialogues », *Artificial Intelligence*, 15(3), 1980.
- [ALL 90] ALLEN J., HENDLER J., TATE A., *Readings in Planning*, Morgan Kaufmann, Representation and Reasoning Series, San Mateo, Californie, 1990.
- [ALT 94] ALTY J.A., GRIFFITHS D., JENNINGS N.R., MAMDANI E.H., STRUTHERS A., WIEGAND M.E., « ADEPT: Advanced decision environment for process tasks: Overview & architecture », dans *Proceedings of the 1994 BCS Expert Systems Conference (Applications Track, ISIP Theme)*, p. 359-371, Cambridge, Grande-Bretagne, 1994.
- [APP 85] APPELT D.E., *Planning English Sentences*, Cambridge University Press, Cambridge, Grande-Bretagne, 1985.
- [BAB 97] BABIN G., MAAMAR Z., CHAIB-DRAA B., « Metadatabase meets distributed AI », dans *First Int. Workshop on Cooperative Information Agents (CIA-97)*, 1997.

- [BAR 95] BARBUCEANU M., FOX M.S., « Cool : a language for describing coordination in multi-agent systems », dans *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-1)*, p. 17-24, San Francisco, Californie, 1995.
- [BAT 92] BATES J., LOYALL A.B., REILLY W.S., An architecture for action, emotion, and social behaviour, Rapport technique CMU-CS-92-144, School of Computer Science, Carnegie-Mellon University, Pittsburgh, Pennsylvanie, mai 1992.
- [BAT 94] BATES J., « The role of emotion in believable agents », *Communications of the ACM*, 37(7), p. 122-125, juillet 1994.
- [BON 88] BOND A.H., GASSER L. (dir.), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, Californie, 1988.
- [BOU 98] BOUZOUBA K., MOULIN B., « Le projet postage : Vers l'amélioration des communications entre agents », *Information In Cognito*, n° 11-12, p. 27-43, 1998.
- [BRA 87] BRATMAN M., *Intention, plans, and practical reason*, Harvard University Press, 1987.
- [BRA 88] BRATMAN M., ISRAEL D., POLLACK M., « Plans and resource bounded practical reasoning », *Computational Intelligence*, 4, p. 349-355, 1988.
- [BRO 86] BROOKS R.A., « A robust layered control system for a mobile robot », *IEEE Journal of Robotics and Automation*, RA-2(1), p. 14-23, 1986.
- [BRO 91a] BROOKS R.A., « Intelligence without representation », *Artificial Intelligence*, vol. 47, n° 1-3, p. 139-159, janvier 1991.
- [BRO 91b] BROOKS R.A., « Intelligence without reason », dans Ray Myopoulos, John Reiter (dir.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, p. 569-595, Morgan Kaufmann, Sydney, Australie, 1991.
- [BUR 96] BURMEISTER B., « Models and methodology for agent-oriented analysis and design », dans K. Fisher (dir.), *Working Notes on the KI'96, Workshop on Agent-Oriented Programming and Distributed Systems*, p. 187-223, DFKI Document, 1996.
- [CAM 83] CAMMARATA S., MCARTHUR D., STEEB R., « Strategies of cooperation in distributed problem solving », dans *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI-83)*, p. 767-770, Karlsruhe, Allemagne, 1983 (repris dans *Readings in Distributed Artificial Intelligence*, Alan H. Bond et Les Gasser (dir.), p. 102-105, Morgan Kaufmann, 1988).
- [CAS 96] CASTELFRANCHI C., CONTE R., « Distributed artificial intelligence and social science: critical issues », dans G. M. P. O'Hare et N. R. Jennings (dir.), *Foundations of Distributed AI*, p. 527-556, John Wiley & Sons, Chichester, Angleterre, 1996.
- [CHA 86] CHAPMAN D., AGRE P., « Abstract reasoning as emergent from concrete activity », dans M. P. George et A. L. Lansky (dir.), *Reasoning About Actions & Plans – Proceedings of the 1986 Workshop*, p. 411-424, Morgan Kaufmann, San Mateo, Californie, 1986.
- [CHA 94a] CHAIB-DRAA B., LEVESQUE P., « Hierarchical models and communication in multi-agent environments », dans *Proceedings of the Sixth European Workshop on*

Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-94), p. 119-134, Odense, Danemark, août 1994.

- [CHA 94b] CHAIB-DRAA B., « Distributed Artificial Intelligence: An overview », dans A. Ken, J. G. Williams, C. M. Hall et R. Kent (dir.), *Encyclopedia Of Computer Science And Technology*, vol. 31, p. 215-243, Marcel Dekker, Inc, 1994.
- [CHA 96a] CHAIB-DRAA B., « Industrial applications of distributed AI », *Communications of the ACM*, vol. 38, n° 11, p. 49-53, 1995.
- [CHA 96b] CHAIB-DRAA B., LEVESQUE P., « Hierarchical model and communication by signs, signals and symbols in multi-agent environments », *Journal of Experimental and Theoretical AI (JETAI)*, 8, p. 7-20, 1996.
- [CHA 96c] CHAIB-DRAA B., « Interaction between agents in routine, familiar and unfamiliar situations », *International Journal of Intelligent and Cooperative Information Systems*, vol. 1, n° 5, p. 7-20, 1996.
- [CHA 99] CHAIB-DRAA B., VANDERVEKEN D., « Agent communication language: towards a semantics based on success, satisfaction and recursion », dans *Agent Theories, Architectures and Language, Intelligent Agents V*, Müller J.P., Singh M.P., Rao A.S.(dir), Springer-Verlag, Lecture Notes on AI, Berlin, 1999.
- [COH 79] COHEN P.R., PERRAULT C.R., « Elements of a plan based theory of speech acts », *Cognitive Science*, 3, p. 177-212, 1979.
- [COH 90a] COHEN P.R., LEVESQUE H. J., « Intention is choice with commitment », *Artificial Intelligence*, 42, p. 213-261, 1990.
- [COH 90b] COHEN P.R., LEVESQUE H.J., « Rational interaction as the basis for communication », dans P. R. Cohen, J. Morgan et M. E. Pollack (dir.), *Intentions in Communication*, p. 221-256, MIT Press, Cambridge, Massachusetts, 1990.
- [COH 95] COHEN P.R., LEVESQUE H.J., « Communicative actions for artificial agents », dans *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS- 1)*, p. 65-72, San Francisco, Californie, 1995.
- [COL 96] COLLINOT A., DROGOUL A., BENHAMOU P., « Agent-oriented design of a soccer robot team », dans *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-2)*, p. 41-47, Kyoto, Japon, 1996.
- [CON 89] CONRY S.E., MEYER R.A., POPE R.P., « Mechanisms for assessing the nonlocal impact of local decisions in distributed planning », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 245-258, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [CON 91a] CONRY S.E., KUWABARA K., LESSER V.R., MEYER R.A., « Multistage negotiation in distributed constraint satisfaction », *IEEE Transactions on Systems, Man, and Cybernetics*, décembre 1991.
- [CON 91b] CONTE R., MICELI M., CASTELFRANCHI C., « Limits and levels of cooperation », dans Y. Demazeau et J.-P. Müller (dir.), *Decentralized AI 2 – Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, p. 147-160, Elsevier Science Publishers B.V., Amsterdam, 1991.

- [COR 83] CORKILL D.D., LESSER V.R., « The use of meta-level control for coordination in a distributed problem solving network », *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, p. 748-756, Karlsruhe, Allemagne, août 1983.
- [COT 98] COTE M., TROUDI N., « Une architecture multi-agent pour la recherche », *L'expertise Informatique*, vol. 3, n° 3, 1998.
- [COT 99] COTE M., Une architecture multi-agent et son application aux systèmes financiers, Master's thesis, Université Laval, Département d'informatique, janvier 1999.
- [DAV 83] DAVIS R., SMITH R., « Negotiation as a metaphor for distributed problem solving », *Artificial Intelligence*, vol. 20, n° 1, p. 63-109, janvier 1983.
- [DEC 89] DECKER K.S., DURFEE E.H., LESSER V.R., « Evaluating research in cooperative distributed problem solving », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 487-519, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [DEM 91] DEMAZEAU Y., MÜLLER J.-P., « From reactive to intentional agents », dans Demazeau Y., Müller J.-P. (dir.), *Decentralized Artificial Intelligence II, Proceedings of the 2nd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-90)*, Elsevier Science Publishes B.V., p. 3-10, 1991.
- [DUR 87] DURFEE E.H., LESSER V.R., « Using partial global plans to coordinate distributed problem solvers », dans *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, Milan, Italie, 1987.
- [DUR 88] DURFEE E.H., *Coordination of distributed problem solvers*, Kluwer, Kluwer international series in engineering and computer science, Boston, 1988.
- [DUR 89] DURFEE E.H., LESSER V.R., « Negotiating task decomposition and allocation using partial global planning », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence Volume II*, p. 229-244, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [ENG 89] ENGELMORE R. S., MORGAN T., *Blackboard Systems*, Addison-Wesley, 1989.
- [ELF 95] EL-FALLAH SEGHRUCHNI A., HADDAD S., « A formal model for coordinating plans in multi-agent systems », dans *Proc. of Intelligent Agents Workshop*, Augusta Technology Ltd, Brooks University, Oxford, Grande-Bretagne, 1995.
- [ELF 96] EL-FALLAH SEGHRUCHNI A., HADDAD S., « A coordination algorithm for multi-agent planning », dans *Proc. of Int. Conf. on Multi-agent Systems-1996 (ICMAS'96)*, AAAI Press, Kyoto, Japon, 1996.
- [FAG 95] FAGIN R., HALPERN I.Y., MOSES Y., VARDI M.Y., *Reasoning about Knowledge*, MIT Press, Cambridge, Massachusetts, 1995.
- [FER 92] FERGUSON I.A., *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*, PhD thesis, Clare Hall, University of Cambridge, Grande-Bretagne, novembre 1992 (repris comme rapport technique n° 273, University of Cambridge, Computer Laboratory).

- [FER 95] FERBER J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, 1995.
- [FIK 71] FIKES R., NILSSON N., « Strips: A new approach to the application of theorem proving to problem solving », *Artificial Intelligence*, vol. 2, n° 3-4, p. 189-208, 1971.
- [FIN 94] FININ T., FRITZSON R., « KQML: a language and protocol for knowledge and information exchange », dans *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence*, p. 126-136, Lake Quinalt, Washington, juillet 1994.
- [FIS 99] FISCHER K., CHAIB-DRAA B., MÜLLER H.J., MÜLLER J.P., PISCHEL M., « A simulation approach based on negotiation and cooperation between agents », *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 29, n° 4, p. 531-545, 1999.
- [FRA 97] FRANKLIN S., GRAESSER A., « Is it an agent, or just a program? A taxonomy for autonomous agents », dans J.P. Mueller, M. Wooldridge et N.R. Jennings (dir.), *Intelligent Agents III: Theories, Architectures, and Languages*, p. 21-35, Springer-Verlag, LNAI 1193, Heidelberg, Allemagne, 1997.
- [GAL 88] GALLIERS J.R., A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict, PhD thesis, Open University, Grande-Bretagne, 1988.
- [GAL 91] GALLIERS J.R., « Cooperative interaction as strategic belief revision », dans S.M. Deen (dir.), *CKBS-90 – Proceedings of the International Working Conference on Cooperating Knowledge Based Systems*, p. 148-163, Springer-Verlag, Heidelberg, Allemagne, 1991.
- [GAS 91] GASSER L., « Social conceptions of knowledge and action: DAI foundations and open systems semantics », *Artificial Intelligence*, n° 47, p. 107-138, 1991.
- [GEO 83] GEORGEFF M.P., « Communication and interaction in multi-agent planning », dans *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, Washington, D.C., 1983.
- [GEO 87] GEORGEFF M.P., LANSKY A.L., « Reactive reasoning and planning », dans *The Proceedings of AAAI-87*, p. 677-682, Seattle, 1987.
- [GEO 98] GEORGEFF M.P., « Communication and interaction in multi-agent planning », dans Bond A.H. et L. Gasser (dir.), *Readings in Distributed Artificial Intelligence*, p. 200-204, Morgan Kaufman, 1998.
- [GLA 96] GLASER N., Contributing to Knowledge Modeling in a Multi-Agent Framework : the CoMoMAS Approach, PhD thesis, Université Henri Poincaré, Nancy, France, 1996.
- [GOO 90] GOODSON J.L., SCHMIDT C.F., « The design of cooperative person-machine problem-solving systems: A methodology and an example », dans W.W. Zachary et S.P. Robertson (dir.), *Cognition, Computation and Cooperation*, p. 187-223, Ablex, 1990.
- [GRA 98] GRAY R., Agent tcl: A flexible and secure mobile-agent systems, Technical Report PCS-TR98-327, Dartmouth College, Computer Science, Hanovier, NH, janvier 1998.

- [HAL 95] HALPERN J.Y., « Reasoning about knowledge: A survey », dans D.M. Gabbay, C.J. Hogger et J.A. Robinson (dir.), *Handbook of Logic in Artificial Intelligence and Logic Programming*, p. 1-34, Oxford University Press, Oxford, Grande-Bretagne, 1995.
- [HAY 98] HAYES-ROTH B., BALL G., LISETTE G., PICARD R., STERN A., « Affect and emotion in the user interface », dans *Proceedings of the International Conference on Intelligent user Interface (IUI-98)*, ACM Press, 1998.
- [HOA 78] HOARE C.A.R., « Communicating sequential processes », *Commun.ACM*, vol. 21, n° 8, p. 666-667, 1978.
- [HUH 87] HUHN M., MUKHOPADHYAY U., STEPHENS L.M., « DAI for document retrieval: The MINDS project », dans M. Huhns (dir.), *Distributed Artificial Intelligence*, p. 249-284, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [IGL 97] IGLESIAS C.A., GARIJO M., GONZÁLEZ J.C., VELASCO J.R., « Analysis and design of multi-agent systems using mas-commonkads », dans *AAAI'97 Workshop on Agent Theories, Architectures and Languages*, Providence, Rhode Island, 1997.
- [JEN 95] JENNINGS N.R., « Controlling cooperative problem solving in industrial multi-agent systems using joint intentions », *Artificial Intelligence*, vol. 74, n° 2, 1995.
- [JEN 98a] JENNINGS N.R., WOOLDRIDGE M., SYCARA K., « A roadmap of agent research and development », *Int. Journal of Autonomous Agents and Multi-Agent Systems*, vol. 1, n° 1, p. 7-38, 1998.
- [JEN 98b] JENNINGS N.R., WOOLDRIDGE M. (dir.), *Agent Technology: Foundations, applications, and Markets*, Springer-Verlag, Heidelberg, Allemagne, 1998.
- [JEN 01] JENNINGS N.R., « An agent-based approach for building complex software system », *Communication of ACM*, vol. 44, n° 4, p. 35-41, 2001.
- [KAE 86] KAEHLING L.P., « An architecture for intelligent reactive systems », dans M.P. Georgeff et A. L. Lansky (dir.), *Reasoning About Actions & Plans – Proceedings of the 1986 Workshop*, p. 395-410, Morgan Kaufmann, San Mateo, Californie, 1986.
- [KAP 98] KAPLAN F., « A new approach to class formation in multi-agent simulation of language evolution », dans *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-2)*, p. 158-165, Paris, 1998.
- [KEN 96] KENDALL E.A., MALKOUN M.T., JIANG C., « A methodology for developing agent-based systems for enterprise integration », dans D. Lukose et Zhang C. (dir.), *First Australian Workshop on Distributed Artificial Intelligence*, p. 216-231, Springer-Verlag, LNAI 1087, 1996.
- [KIN 96] KINNY D., GEORGEFF M., RAO A., « Methodology and modelling technique for systems of bdi agents », dans W. van der Welde et J. Perram (dir.), *Decentralized AI 2 – Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, Springer-Verlag, LNAI 1038, 1996.
- [KOB 89] KOBSA A., *User Models in Dialog Systems*, Springer-Verlag, Heidelberg, Allemagne, 1989.

- [KRA 91] KRAUS S., WILKENFELD J., « Negotiations over time in a multi-agent environment », dans *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, p. 56-61, Sydney, Australie, 1991.
- [KRA 95] KRAUS S., WILKENFELD J., ZLOTKIN G., « Multi-agent negotiation under time constraints », *Artificial Intelligence*, vol. 75, n° 2, p. 297-345, 1995.
- [KRE 91] KREIFELTS T., VON MARTIAL F., « A negotiation framework for autonomous agents », dans Y. Demazeau et J.-P. Müller (dir.), *Decentralized AI 2 – Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, p. 71-88, Elsevier Science Publishers B.V., Amsterdam, 1991.
- [LAK 99] LAKEMEYER G., LEVESQUE H., « Query evaluation and progression in AOL knowledge bases », dans *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, p. 124-131, Stockholm, Suède, 1999.
- [LES 80] LESSER V.R., ERMAN L.D., « Distributed interpretation: A model and experiment », *IEEE Transactions on Computers*, C-29, n° 12, 1980.
- [LES 91a] LESPÉRANCE Y., A Formal Theory of Indexical Knowledge and Action, PhD thesis, Toronto, 1991.
- [LES 91b] LESSER V.R., « A retrospective view of FA/C distributed problem solving », *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, vol. 21, n° 6, p. 1347-1362, 1991.
- [LES 94] LESPÉRANCE Y., LEVESQUE H.J., LIN F., MARCU D., REITER R., SCHERL R.B., « A logical approach to high-level programming – A progress report », dans Benjamin Kuipers (dir.), *In Control of the Physical World by Intelligent Systems, Papers from the 1994 AAAI fall Symposium*, p. 78-85, La Nouvelle-Orléans, Louisiane, 1994.
- [LEV 90] LEVESQUE H.J., NUNES J.H.T., COHEN P.R., « On acting together », dans William Dietterich et Tom Swartout (dir.), *Proceedings of the 8th National Conference on Artificial Intelligence*, p. 94-99, Hynes Convention Centre, MIT Press, 1990.
- [LIT 87] LITMAN D.J., ALLEN P.R., « A plan recognition model for subdialogues in conversations », *Cognitive Science*, 11, p. 163-200, 1987.
- [MAA 97] MAAMAR Z., MOULIN B., « An agent based approach for intelligent and cooperative systems », dans *Proc. of IEEE Knowledge and Data Engineering Exchange Workshop'97 (KDEX-97)*, p. 41-47, Newport Beach, Californie, 1997.
- [MAE 94a] MAES P., « Agents that reduce work and information overload », *Communications of the ACM*, vol. 37, n° 7, p. 31-40, juillet 1994.
- [MAE 94b] MAES P., « Social interface agents: Acquiring competence by learning from users and other agents », dans O. Etzioni (dir.), *Software Agents – Papers from the 1994 Spring Symposium (Technical Report SS-94-03)*, p. 71-78, AAAI Press, mars 1994.
- [MAL 90] MALONE T.W., « Organizing information processing systems: parallels between human organizations and computer systems », dans W.W. Zachary et S.P. Robertson (dir.), *Cognition, Computation and Cooperation*, p. 56-83, Ablex, 1990.

- [MAL 98] MALVILLE E., BOURDON F., « Task allocation: A group self-design approach », dans *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS- 2)*, p. 166-167, Kyoto, Japon, 1998.
- [MIN 79] MINTZBERG H., *The Structuring of Organizations*, Prentice Hall, Englewoods Cliffs, 1979.
- [MOU 94] MOULIN B., CLOUTIER L., « Collaborative work based on multi-agent architectures: a methodological perspective », dans F. Aminzadeh et M. Jamshidi (dir.), *Artificial Intelligence: Theory and Applications*, p. 187-223, Prentice Hall, Englewoods Cliffs, 1994.
- [MOU 96] MOULIN B., CHAIB-DRAA B., « An overview of distributed artificial intelligence », dans G.M.P. O'Hare et N.R. Jennings (dir.), *Foundations of Distributed AI*, p. 3-54, John Wiley & Sons, Chichester, Grande-Bretagne, 1996.
- [MUL 96] MÜLLER J., « Negotiation principles », dans G.M.P. O'Hare et N.R. Jennings (dir.), *Foundations of Distributed AI*, p. 211-230, John Wiley & Sons, Chichester, Grande-Bretagne, 1996.
- [NEG 95] NEGROPONTE A., *Being Digital*, Hodder and Stoughton, 1995.
- [NEW 61] NEWELL A., SIMON H.A., « GPS, a program that simulates human thought », dans Heinz Billing (dir.), *Lernende Automaten*, p. 109-124, R. Oldenbourg, Munich, Allemagne, 1961.
- [NII 89a] NII P., « Blackboard systems », dans *Handbook of Artificial Intelligence*, vol. IV, Addison-Wesley, Reading, Massachusetts, 1989.
- [NII 89b] NII P., AIELLO N., RICE J., « Experiments on Cage and Polygon: Measuring the performance of parallel blackboard systems », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 319-384, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [NOD 98] NODIN M., BOHRER W., HIONG NGU A., Semantic brokering over dynamic heterogeneous data sources in infosleuth, Technical Report, MCC Corporation, 1998.
- [NWA 99] NWANA H.S., NDUMU D.T., « Agents of change in future communication systems », *Applied AI Journal*, vol. 13, n° 1, 1999.
- [RAO 91a] RAO A.S., GEORGEFF M.P., « Asymmetry thesis and side-effect problems in linear time and branching time intention logics », dans *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, p. 498-504, Sydney, Australie, 1991.
- [RAO 91b] RAO A.S., GEORGEFF M.P., « Modeling rational agents within a BDI-architecture », dans R. Fikes et E. Sandewall (dir.), *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, p. 473-484, Morgan Kaufmann, San Mateo, Californie, avril 1991.
- [RAO 92] RAO A.S., GEORGEFF M.P., « An abstract architecture for rational agents », dans William Nebel, Bernhard Rich, Charles Swartout (dir.), *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, p. 439-449, Cambridge, Massachusetts, Morgan Kaufmann, 1992.

- [ROS 85a] ROSENSCHEIN S.J., « Formal theories of knowledge in AI and robotics », *New Generation Computing*, p. 345-357, 1985.
- [ROS 85b] ROSENSCHEIN S.J., Rational Interaction: Cooperation Among Intelligent Agents, PhD thesis, Stanford University, Computer Science Department, 1985.
- [ROS 86] ROSENSCHEIN S.J., GINSBERG M., GENESERETH M.R., « Cooperation without communication », dans *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphie, Pennsylvanie, 1986.
- [ROS 89] ROSENSCHEIN S.J., BREESE J.S., « Communication-free interactions among national agents », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 99-118, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [ROS 94] ROSENSCHEIN S.J., ZLOTKIN G., *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computer*, MIT Press, Boston, Massachusetts, 1994.
- [ROT 89] ROTH B.H., HEWETT M., WASHINGTON R., HEWETT R., SEIVER A., « Distributing intelligence within an individual », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. 2, p. 385-412, Pitman/Morgan Kaufmann, Londres/San mateo, 1989.
- [ROU 95] ROUSSEAU D., MOULIN G., LAPALME B., « Cool: a language for describing coordination in multi-agent systems », dans *Actes des Deuxièmes journées francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, p. 3-14, Voiron, France, 1995.
- [RUS 96] RUS D., SUBRAMANIAN D., Information retrieval, information structure, and information agents, Rapport Technique PCS-TR96-255, Dartmouth College, Computer Science, Hanovier, NH, janvier 1996.
- [SAN 95] SANDHOLM T., LESSER V., « Issues in automated negotiation and electronic commerce: Extending the contract net framework », dans *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-1)*, p. 328-335, San Francisco, Californie, 1995.
- [SEA 69] SEARLE J.R., *Speech Acts: An Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, Grande-Bretagne, 1969.
- [SHO 92] SHOHAM Y., TENNENHOLTZ M., « On the synthesis of useful social laws for artificial agent societies », dans *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Diego, Californie, 1992.
- [SHO 93] SHOHAM Y., « Agent-oriented programming », *Artificial Intelligence*, vol. 60, n° 1, p. 51-92, 1993.
- [SIN 94] SINGH M.P., *Multi-agent Systems: A Theoretical Framework for Intentions, Know-How, and Communications*, Springer-Verlag, LNAI 799, Heidelberg, Allemagne, 1994.
- [SMI 80a] SMITH R.G., « The contract net protocol », *IEEE Transactions on Computers*, vol. C-29, n° 2, 1980.
- [SMI 80b] SMITH R.G., DAVIS R., « Frameworks for cooperation in distributed problem solving », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, n° 1, 1980.

- [SMI 94] SMITH D.C., CYPHER A., SPOHRER J., « KIDSIM: Programming agents without a programming language », *Communications of the ACM*, vol. 37, n° 7, p. 55-67, juillet 1994.
- [SYC 88] SYCARA K.P., « Resolving goal conflicts via negotiation », dans *Proceedings of the 7th National Conference on Artificial Intelligence*, p. 245-250, St-Paul, Minnesota, 1988.
- [SYC 89] SYCARA K.P., « Multi-agent compromise via negotiation », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 119-138, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [SYC 90a] SYCARA K.P., « Negotiation planning: An AI approach », *European Journal of Operational Research*, 46, p. 216-234, 1990.
- [SYC 90b] SYCARA K.P., « Persuasive argumentation in negotiation », *Theory and Decision*, 28, p. 203-242, 1990.
- [THO 93] THOMAS S.R., PLACA, an Agent Oriented Programming Language, PhD thesis, Stanford University, Computer Science Department, Stanford, CA 94305, août 1993.
- [TRO 99] TROUDI N., Un système multi-agent pour les environnements riches en information, Master's thesis, Université Laval, Département d'informatique, avril 1999.
- [TUB 84] TUBBS S.L. (dir.), *A system Approach to Small Group Interaction*, 2^e édition, Addison Wesley, Reading, Massachusetts, 1984.
- [VER 97] VERHAREN. E.M., A Language-Action Perspective on the Design of Cooperative Information Agents, PhD thesis, Katholieke Universiteit Brabant, Pays-Bas, 1997.
- [VID 98] VIDAL M.J., MÜLLER T., WEINSTEIN P., DURFEE H.E., « The university of michigan digital library service market society », dans K. P. Sycara et M. Wooldridge (dir.), *Proc. of the 2nd Int. Conf. on Autonomous Agent (Agent-98)*, p. 269-276, 1998.
- [VMA 90] MARTIAL F.V., « Interactions among autonomous planning agents », dans Y. Demazeau et J.-P. Müller (dir.), *Decentralized AI – Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*, Elsevier Science Publishers B.V., Amsterdam, 1990.
- [VMA 92] MARTIAL F.V. (dir.), *Coordinating Plans of Autonomous Agents*, Springer-Verlag, Heidelberg, Allemagne, 1992.
- [VON 99] VONGKASEM L., CHAIB-DRAA B., « Acl as a joint project between participants: A preliminary report », dans *First Workshop on Agent Communication Language (ACL'99)*, Stockholm, Suède, 1999.
- [VPA 96] PARUNAK H.V.D., « Applications of distributed artificial intelligence in industry », dans G.M.P. O'Hare et N.R. Jennings (dir.), *Foundations of Distributed AI*, John Wiley & Sons, Chichester, Grande-Bretagne, 1996.
- [WAV 95] WAVISH P., GRAHAM M., « Roles, skills, and behaviour: a situated action approach to organising systems of interacting agents », dans *Intelligent Agents: theories, Architectures, and Languages*, p. 371-385, Springer-Verlag, LNAI 890, Heidelberg, Allemagne, 1995.

- [WER 89] WERNER E., « Cooperating agents: A unified theory of communication and social structure », dans L. Gasser et M. Huhns (dir.), *Distributed Artificial Intelligence*, vol. II, p. 3-36, Pitman Publishing/Morgan Kaufmann, Londres/San Mateo (Californie), 1989.
- [WER 92] WERNER E., « The design of multi-agent systems », dans E. Werner et Y. Demazeau (dir.), *Decentralized AI 3 – Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, p. 3-30. Elsevier Science Publishers B.V., Amsterdam, 1992.
- [WIL 88] WILKINS D.E., *Practical Planning: Extending the AI Planning Paradigm*, Morgan Kaufmann, San Mateo, Californie, 1988.
- [ZEN 96] ZENG D., SYCARA K., « Coordination of multiple intelligent software agents », *International Journal of Cooperative Information Systems*, n° 5, p. 181-212, 1996.

Chapitre 2

Modèles et architectures d'agents

2.1. Introduction

L'intelligence artificielle distribuée (IAD) et les systèmes multi-agents (SMA) ont introduit le terme d'*agent* dans le domaine de l'informatique pour désigner les entités logicielles ou physiques interagissant dans les systèmes qu'ils construisent.

Au fil des avancées théoriques et des applications, de multiples définitions de ce terme sont apparues¹. Ainsi, pour certains, il englobe des capacités de planification, pour d'autres d'apprentissage ou encore de mobilité, etc. Afin de fixer le cadre de ce chapitre, nous prendrons une définition minimale² que nous enrichirons par la suite : un *agent* est un programme informatique *autonome*, plongé dans un *environnement* qu'il est capable de *percevoir* et sur lequel il est capable d'*agir*. Nous définirons rapidement l'*environnement*³ comme le médium commun aux agents du système. L'agent peut le contrôler totalement, ou partiellement, au travers de ses actions, à partir d'une perception locale, incomplète, voire incorrecte, de celui-ci. Le

1. Le lecteur intéressé pourra se reporter à [FRA 96, LUC 95] pour une revue de différentes définitions du terme *agent*.

2. Cette définition sera cependant plus large que celle énoncée dans le premier chapitre de cet ouvrage puisqu'elle intègre les agents réactifs.

3. Un environnement peut être indéterministe, dynamique, continu, etc. (se reporter à [RUS 95] pour une définition plus complète).

qualificatif d'*autonomie*⁴ est essentiel pour distinguer un agent d'un objet ou de tout autre programme informatique. Le comportement d'un agent peut être qualifié ainsi, lorsqu'il peut exercer son propre contrôle sur son activité et sur son état.

Comme le montrent les autres chapitres de cet ouvrage, la conception d'agents occupe une place importante au sein des recherches en SMA. Les caractéristiques de l'environnement, les types d'interactions possibles avec les autres agents, les modes de coordination influent sur le ou les type(s) d'agent(s) à utiliser. Les concepteurs actuels sont donc confrontés au difficile choix du type d'agent étant données les caractéristiques de leur application. L'objectif de ce chapitre est de présenter un panorama des modèles d'agent existants. Afin de limiter notre étude, nous nous intéressons essentiellement aux agents opérant dans un *univers multi-agent*⁵. Les travaux que nous avons retenus sont soit à l'état de maquette ou de prototype, issus directement du domaine de la recherche, soit à l'état de produits commercialisés. Nous précisons, dans un premier temps, le niveau de description retenu pour ce panorama. Nous élaborerons ensuite la grille de classification selon laquelle nous avons choisi d'organiser les types de modèles d'agent existants. Nous aurons alors en main tous les éléments pour parcourir les travaux réalisés en nous arrêtant sur ceux qui sont représentatifs de la problématique illustrée. Nous conclurons par un tableau récapitulatif des architectures selon les dimensions évoquées et donnerons quelques perspectives.

2.2. Modèles d'agent

Les modèles d'agent sont nombreux dans la littérature. Au-delà des terminologies utilisées, des domaines d'application, du niveau d'abstraction et des aspects mis en avant, leurs descriptions diffèrent par le type de représentation des connaissances (uniforme/hétérogène, implicite/explicite, symbolique/numérique), par le type de contrôle (centralisé/hierarchique, concurrent/séquentiel) mis en œuvre. Dans ce chapitre, nous nous attacherons à une description *interne*⁶ telle que le *concepteur* a pu la définir.

4. Les définitions de l'autonomie sont multiples dans le domaine, se reporter à [CAS 95, CAS 98a, CAS 98b] pour plus de détails et pour une définition plus large que celle que nous utilisons ici.

5. Nous laissons ainsi de côté, notamment, les agents mobiles [MIL 99], les agents d'interface [GEN 94] ou ceux dotés uniquement de capacités d'apprentissage.

6. Dans les travaux sur les SMA, d'autres descriptions d'agents existent. Le lecteur intéressé pourra notamment se reporter à [DEM 91a] où la distinction entre description *externe* et *interne* d'agents est faite ou aux recherches portant sur la représentation des autres [GMY 95, SIC 95, VER 00] au sein d'un SMA.

2.2.1. Architecture/comportement d'agent

Nous appelons *architecture* d'un agent, la structure logicielle (ou matérielle) qui, à partir d'un ensemble d'entrées, produit un ensemble d'actions sur l'environnement ou sur les autres agents. Sa description [SHA 95] est constituée des composants (correspondant aux fonctions) de l'agent et des interactions entre ceux-ci (flux de données et de contrôle).

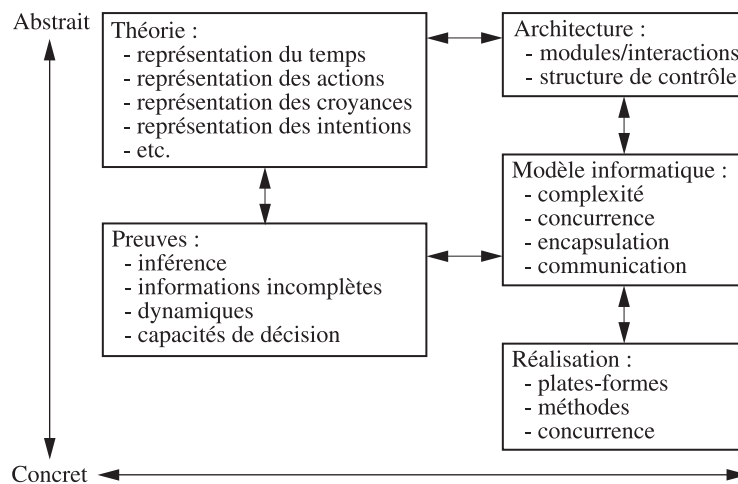


Figure 2.1. Espace de conception et de description des agents (d'après [JUN 99])

Au-delà de cette description architecturale, la nature des agents conduit naturellement à les modéliser comme des machines intentionnelles [BRA 87, DEN 87]. De ce fait, certaines descriptions concernent la définition des *attitudes mentales* manipulées ainsi que leurs relations. Ces attitudes peuvent être *épistémiques* telles que *croyances*, *connaissances* et/ou *conatives* telles que *désirs* (ce que l'agent veut faire), *intentions* (décisions d'action prises pour satisfaire les désirs étant données les croyances), *obligations*, *engagements*. De nombreuses *théories agents* ont été développées afin de définir formellement ces attitudes et leurs relations. Le qualificatif d'agent attaché à celles-ci vise à préciser leur orientation principale : formaliser et spécifier les attitudes d'un type d'agent donné.

Les théories agents⁷ les plus connues sont le modèle *Belief Desire Intention* [BRA 88, GEO 87, GEO 99], modèle d'agent rationnel [COH 90]. Différentes logiques [FAG 95, MOO 90, RAO 91, SIN 94] supportent les langages formels et déclaratifs sur lesquels elles s'appuient. Nous ne les décrivons pas dans le cadre de ce chapitre.

2.2.2. Descriptions d'un agent

La conception d'un agent, comme toute activité logicielle, peut et doit se dérouler à différents niveaux de conceptualisation détaillant successivement la description (voir figure 2.1) en *spécification*, *conception* et *implémentation*. Orthogonale à ces niveaux d'abstraction, la description peut être *déclarative* en s'appuyant sur des théories agents ou *opérationnelle* et mettre en avant la description de l'architecture.

Au *niveau spécification*, la description architecturale cherche à mettre en évidence les composants qui constituent l'agent, les fonctions qu'ils mettent en œuvre, leurs interactions avec les autres composants, leur organisation globale. La ou les description(s) formelle(s) spécifie(nt) le comportement désiré en précisant les différentes représentations disponibles, les attitudes qui composent l'état de l'agent, si une représentation explicite ou non du temps est présente, etc. A ce niveau, les deux types de descriptions ne sont pas satisfaisants de par leur manque de précision (description architecturale) ou à cause de leur manque de caractère opérationnel (description logique).

Au *niveau conception*, les descriptions architecturales sont affinées en précisant le grain des traitements, les modèles de concurrence à utiliser, la répartition des processus, etc. Certains travaux utilisent des langages de spécification formelle tels que, par exemple, le langage Z [GOO 93, INV 97, RAO 95] ; d'autres ont utilisé les automates cellulaires [FER 97], les réseaux de Petri [FER 95a]. Par ailleurs, afin d'exploiter les spécifications logiques, des mécanismes d'inférence et de preuve sont définis et mis en œuvre.

Au niveau le plus bas se trouve la description la plus concrète et la plus opérationnelle : l'implémentation de l'agent dans un langage donné. Ce niveau se prête peu aux comparaisons. Des langages objets ou acteurs, des langages de programmation orientés agents [RAO 96, THO 95] ou des environnements de développement multi-agent (voir chapitre 5) pourront être utilisés.

7. Le lecteur intéressé pourra se reporter à [RAO 99, WOO 99] pour une présentation plus exhaustive de celles-ci.

Les agents que l'on trouve dans la littérature ne possèdent pas tous toutes ces descriptions. Souvent leurs propriétés ne sont pas clairement formalisées selon une des théories citées ci-dessus, mais sont simplement exprimées au sein du code. Avec tout l'implicite qui peut en découler, il est difficile de les vérifier. Ainsi, un des problèmes de la conception d'agents est lié à l'absence d'un formalisme unique permettant de passer de la spécification à l'implémentation en étant sûr de conserver les propriétés désirées de l'agent (calcul de preuves, etc.). Ces descriptions formelles sont souvent trop abstraites pour dériver des programmes exécutables. Elles peuvent être cependant utilisées pour vérifier des propriétés de l'agent dans des conditions restrictives [FIS 97, RAO 93, WOO 92]. Les implantations existantes des architectures BDI, par exemple, s'éloignent beaucoup de la formalisation logique et de la théorie BDI notamment sur la gestion de l'évolution des désirs, des croyances et des intentions, avec les liens qui peuvent exister entre eux. Des tentatives sont cependant réalisées (proposition, par exemple, dans [RAO 92], d'une architecture abstraite rendant opérationnelle la théorie BDI). Dans un autre ordre d'idée, le langage MetateM [FIS 97] fondé sur une logique temporelle permet de spécifier des agents et d'exécuter directement cette spécification. Le travail réalisé autour des automates situés va également dans ce sens [ROS 86] avec l'élaboration d'un langage de spécification déclarative d'un agent qui peut être compilé en un circuit logique.

Dans la suite de ce chapitre, nous nous situerons au niveau *spécification* en considérant une description *architecturale* de l'agent. Au-delà de cette description, nous essaierons de mettre également en avant les objectifs et les motivations qui ont donné lieu au développement de l'architecture et les domaines d'application dans lesquels elle a été utilisée.

2.3. Dimensions des architectures d'agents

Nous allons maintenant décrire la méthodologie qui a prévalu pour l'étude et la description des architectures d'agent de ce chapitre. Nous avons axé notre analyse sur les capacités disponibles au sein d'un agent par rapport à la problématique centrale des SMA que constitue la *coordination* entre agents (voir chapitre 4). Nous obtenons ainsi une grille principale organisant les descriptions d'agent en architectures *d'agents autonomes*, architectures *d'agents interagissants* ou architectures *d'agents sociaux*. Afin d'affiner notre analyse, au sein de chacune de ces rubriques, nous avons pris en compte le couplage de l'agent à l'environnement et la manière dont la fonction de prise de décision est réalisée. Nous retrouvons ainsi les classifications habituelles [HUH 97, MUL 97, WOO 99], d'architecture *d'agent réactif*, *d'agent délibératif* ou *d'agent hybride*.

2.3.1. Vision coordination

Afin de préciser les capacités liées à la coordination, nous nous appuyons sur l'approche *VOYELLES* [DEM 95] qui modélise un SMA (niveau macro) selon quatre dimensions qui sont : *Agent (A)*, *Environnement (E)*, *Interaction (I)* et *Organisation (O)*. Au niveau micro (celui de l'agent, entité agissante du système), celles-ci se retrouvent au sein de l'architecture de l'agent (voir figure 2.2), déclinées en quatre facettes explicites ([BOI 93, BOU 92] par exemple) ou combinées entre elles au sein des composants de l'architecture de l'agent⁸. La facette *A* désigne l'ensemble des fonctionnalités de raisonnement interne de l'agent (planification, introspection, par exemple). La facette *E* regroupe les fonctionnalités liées aux capacités de perception et d'action de l'agent sur l'environnement. La facette *I* regroupe les fonctionnalités liées aux capacités d'interaction de l'agent avec les autres agents (interprétation des primitives du langage de communication, gestion des protocoles d'interaction et des conversations par exemple, voir chapitre 3). La facette *O* est peut-être la plus difficile à définir. Elle concerne les fonctions et les représentations liées aux capacités de structuration et de gestion des relations des agents entre eux⁹.

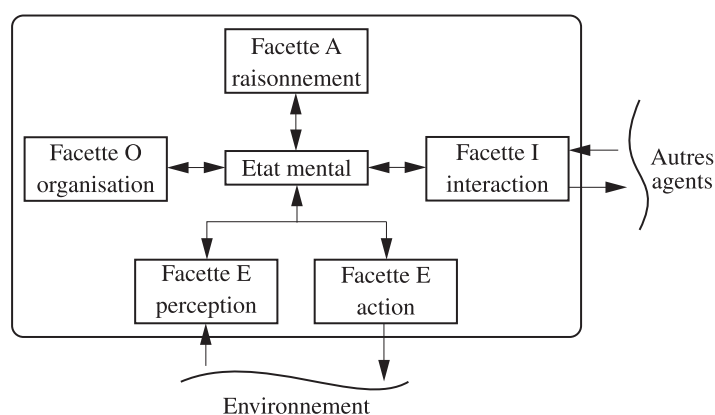


Figure 2.2. Facettes AEIO au sein d'un agent

8. C'est pourquoi nous préférons les appeler des *facettes* plus que des composants.

9. Cette structuration peut s'exprimer en définitions de statuts et de comportements sociaux [JEN 93a, PAT 87, SHO 95], de rôles et de relations [CAV 98, HAN 99] (subordination), de dépendances [CAS 92, SIC 95], d'engagements [BOU 92]), de groupes [BAE 98], etc.

La présence ou l'absence de ces facettes conduit à distinguer différents types d'architectures. A partir de la définition minimale donnée en introduction, nous ne considérerons pas les architectures en possession uniquement de la facette *A*. Nous retrouverions les architectures cognitives de l'intelligence artificielle (IA) sans connexion avec un environnement et raisonnant ainsi dans un monde clos. Nous sommes ainsi amenés à considérer les trois types d'architecture suivants.

2.3.1.1. *Architecture d'agent autonome*

Nous appelons *architecture d'agent autonome*¹⁰ l'architecture d'un agent, possédant des capacités d'action et de perception sur son environnement. Cette architecture est ainsi constituée des fonctions exprimant les facettes *A* et *E*. L'agent possédant une telle architecture répondra ainsi à la définition minimale donnée en introduction, sans capacité explicite de coopération avec d'autres agents.

2.3.1.2. *Architecture d'agent interagissant*

Nous appelons *architecture d'agent interagissant*, l'architecture d'un agent possédant des capacités d'interaction avec les autres agents du système. L'agent possédera éventuellement des descriptions des autres agents et sera capable de les influencer *via* des envois de messages. Une telle architecture est ainsi constituée des fonctions correspondant aux facettes *A*, *I* et *E*.

La possession de la facette *E* est optionnelle : c'est ce qui se produit pour certains agents dont les capacités d'action dans l'environnement sont nulles mais qui possèdent des capacités de communication avec les autres agents.

2.3.1.3. *Architecture d'agent social*

Le passage au niveau supérieur (agent interagissant à agent social) fait intervenir des mécanismes de coordination d'un niveau supérieur tels que la gestion et la génération d'organisations : des entités organisationnelles ou équipes sont formalisées et prises en compte par l'agent pour entamer des coopérations. Nous appelons, *architecture d'agent social*, l'architecture d'un agent possédant des capacités d'interaction avec les autres agents du système et les capacités de gestion des relations entretenues avec les autres agents du système. Une telle architecture est ainsi constituée des fonctions correspondant aux facettes *A*, *I*, *O* et *E*, cette dernière facette étant facultative.

Notons que cette définition diffère de son acception courante dans la littérature [CAS 98a, MOU 96]. En effet, le qualificatif de social est utilisé souvent pour

10. Le nom de ce type d'architecture est choisi en référence au terme *autonomous agent* utilisé en robotique ou dans certains domaines [MAE 90a].

désigner la capacité qu'offre l'architecture à l'agent, de coopérer avec d'autres agents dans le système. Nous le particularisons ici à la capacité de gestion d'organisations.

2.3.2. *Vision raisonnement*

Tel que nous l'avons défini dans l'introduction, un agent peut être considéré comme un système de raisonnement visant à déterminer *quelles* sont les actions possibles, *laquelle* privilégier, *comment* la mettre en œuvre de manière efficace en fonction de la situation courante de l'environnement ou des autres agents. Trois grands groupes d'architectures permettent de répondre différemment à ces questions.

2.3.2.1. *Architecture d'agent réactif*

Une *architecture d'agent réactif* est une architecture dans laquelle les décisions d'action sont prises au moment de l'exécution à partir de peu d'informations (informations captées à l'instant présent par exemple). On assiste à un couplage direct de l'action à la perception. L'exécution d'une action suit immédiatement un changement intervenant dans l'environnement. Souvent, pour faciliter ce couplage, ces architectures n'incluent aucun type de représentation symbolique de l'environnement. Le mécanisme de décision est souvent très simple, l'accent étant mis sur la robustesse plutôt que sur l'exactitude et l'optimum du comportement.

2.3.2.2. *Architecture d'agent délibératif*

Une *architecture d'agent délibératif* comporte un mécanisme de décision consistant en une *délibération* explicite sur les différentes possibilités d'action en utilisant, par exemple, une génération de plans d'action [ALL 90], ou en prenant en compte l'utilité espérée des actions [RUS 91]. Découlant de ce mécanisme de décision complexe, ces architectures possèdent une représentation explicite et symbolique de l'environnement, à partir de laquelle le raisonnement peut déterminer quelles actions peuvent être exécutées dans l'environnement. Le mécanisme de décision pouvant être très complexe, l'accent est mis ici sur l'optimum et sur l'exactitude du comportement au détriment de la robustesse et du temps d'exécution.

2.3.2.3. *Architecture d'agent hybride*

Certaines recherches se sont intéressées au développement d'architectures permettant de concilier les avantages sans les inconvénients de chacune des deux approches précédentes. Une *architecture d'agent hybride* est le résultat de la combinaison d'architectures d'agent délibératif et d'agent réactif. Leur objectif est d'obtenir un comportement qui soit cohérent avec les buts et les plans de l'agent, dont les réactions soient guidées par les objectifs et non plus seulement par les stimuli provenant de l'environnement.

Dans ce qui suit, nous allons utiliser les deux grilles que nous venons de définir. Ces deux visions orthogonales nous conduiront à distinguer trois grandes familles d'agents (autonomes, interagissants, sociaux) au sein desquelles nous distinguerons les agents réactifs, délibératifs et hybrides.

2.4. Architectures d'agents autonomes

Se démarquant des architectures d'agents cognitifs développées en IA, les architectures d'agents autonomes ont été les premières à introduire le couplage de l'agent avec l'environnement dans lequel il est plongé. De nombreuses applications sont issues de la robotique.

2.4.1. Architectures d'agents réactifs

Les architectures d'agents réactifs autonomes sont nombreuses. Elles se distinguent selon la liaison entre perception et action. Certaines architectures [CHA 90, KAE 86, ROS 86, WAV 92] organisent les traitements autour des situations que l'on peut trouver dans l'environnement. Dans PENG1 [AGR 87], par exemple, des règles relient des configurations particulières de l'environnement à des actions à exécuter. D'autres architectures, comme [BRO 86] ou MANTA [DRO 93], que nous allons détailler par la suite, ont préféré privilégier les comportements élémentaires [MAE 90b, STE 89] reliés à un ensemble de stimuli. D'autres, enfin, comme PACO [DEM 93], que nous présenterons dans le cadre des architectures *d'agents interagissants*, ont privilégié les interactions élémentaires entre l'agent et l'environnement ou les autres agents.

2.4.1.1. MANTA

Pour beaucoup de systèmes réactifs, la simulation est un domaine d'application privilégié (voir chapitre 7). Le système MANTA (*Modeling an ANThill Activity*) [DRO 93] a été utilisé dans le domaine de la simulation éthologique. Dans ce système, l'architecture d'un agent (voir figure 2.3) comporte les opérations de *perception*, de *sélection* et d'*activation* qui manipulent un ensemble de *tâches*. Une tâche est une séquence d'actions primitives directement exécutables par les effecteurs de l'agent dans l'environnement. Elle est caractérisée par un *poids* (importance de la tâche au sein de l'agent), un *seuil* et un *niveau d'activation*. Après chaque exécution d'une action primitive de la tâche courante, l'opération de *perception* met en relation les stimuli issus de l'environnement avec les tâches correspondantes ; la *sélection* calcule le niveau d'activation de chacune des tâches en fonction de leur poids, de l'intensité des stimuli et choisit les tâches dont le niveau d'activation est supérieur à leur seuil. La tâche activée par l'opération d'activation sera celle dont le niveau d'activation est le plus élevé parmi les tâches

choisies et la tâche courante : si la tâche courante est prolongée, son niveau d'activation est décrémenté, sinon le poids de la tâche activée est incrémenté et l'ancienne tâche est suspendue. On voit ainsi un mécanisme de *sélection par compétition* des tâches adapté en fonction de l'activité courante, ce qui permet d'éviter un comportement chaotique. Au niveau du système, cette régulation a notamment permis de mettre en évidence l'émergence de spécialisations et de répartition des tâches entre les agents.

2.4.1.2. Architecture Subsumption

Dans les années 1980, Brooks s'est élevé contre les thèses de l'école cognitive en défendant l'idée qu'un système pouvait être intelligent sans manipuler d'informations symboliques [BRO 91]. L'architecture *Subsumption* qu'il a proposée repose sur cette thèse. Elle a été utilisée dans le domaine de la robotique [BRO 86, NAK 98, STE 90].

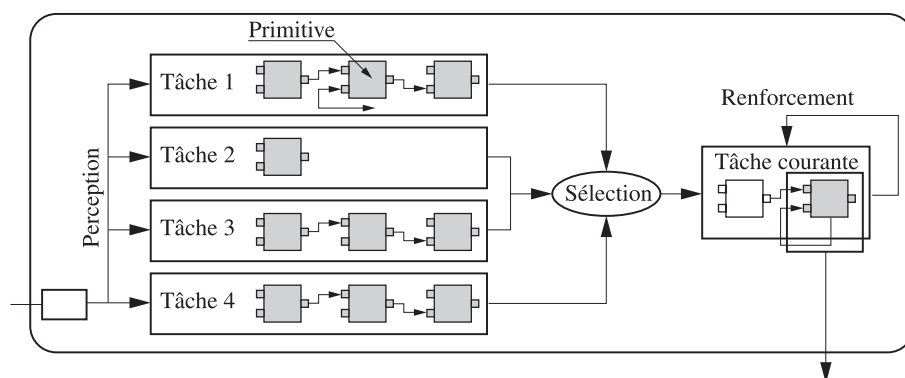


Figure 2.3. Architecture MANTA

Cette architecture d'agent réactif (voir figure 2.4) intercale entre la perception et l'action, différentes couches de comportement de complexité croissante : sur la couche basse, les comportements de base tels que le mouvement aléatoire ; sur les couches les plus hautes ceux devant permettre, théoriquement, de satisfaire des comportements spécifiques. Les couches fonctionnent en parallèle et accèdent à un sous-ensemble des informations captées. L'accès au système d'action est contrôlé *via* une hiérarchie de contrôle : les comportements présents sur une couche peuvent supplanter les comportements présents sur les couches inférieures en substituant leurs propres entrées aux entrées des couches inférieures et en inhibant toute sortie de ces mêmes couches pour des périodes de temps préprogrammées. A part ces signaux descendants de suppression et d'inhibition, aucun autre type de communication n'a lieu entre les

couches. Toute la difficulté de conception d'un tel agent réside dans la définition de la hiérarchie de contrôle entre couches. Des applications ont ainsi pu être réalisées pour des tâches simples. A notre connaissance, aucune application nécessitant une planification ou une coopération avec d'autres agents n'a été réalisée.

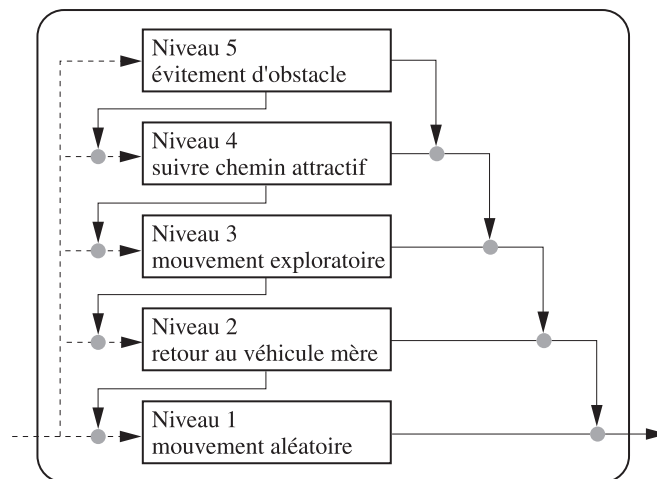


Figure 2.4. Architecture Subsumption d'après [STE 90]

A la différence de l'approche précédente dans MANTA et de nombreuses applications réactives, qui plaçaient l'intelligence au niveau du système multi-agent comme le résultat de l'interaction entre des agents simples, l'intelligence est placée ici au niveau de l'organisation des comportements au sein de l'agent.

2.4.2. Architectures d'agents délibératifs

De nombreuses architectures d'agents délibératifs ont été élaborées. Ce sont notamment celles qui étendent des architectures cognitives de l'IA telles que SOAR (*State OperAtor Result*) [NEW 90], AIS (*Adaptative Intelligent System*) [HAY 95] s'appuyant sur l'architecture de tableau noir *BB1* [HAY 85].

Procedural Reasoning System

Nous avons choisi de décrire plus précisément l'architecture PRS/dMARS (*Procedural Reasoning System*) [GEO 87, GEO 89] pour son impact actuel dans le domaine. Elle appartient à la famille des architectures BDI dont un précurseur était le système IRMA [BRA 88]. Elle a fait l'objet de nombreuses évolutions tant d'un

point de vue théorique que pratique (version plus récente et aboutie avec dMARS [INV 97], langage de programmation AgentSpeak(L) [RAO 96], etc.). Elle a été appliquée dans différents domaines (gestion de *workflows*, supervision de réseaux de télécommunication, robotique mobile, etc.).

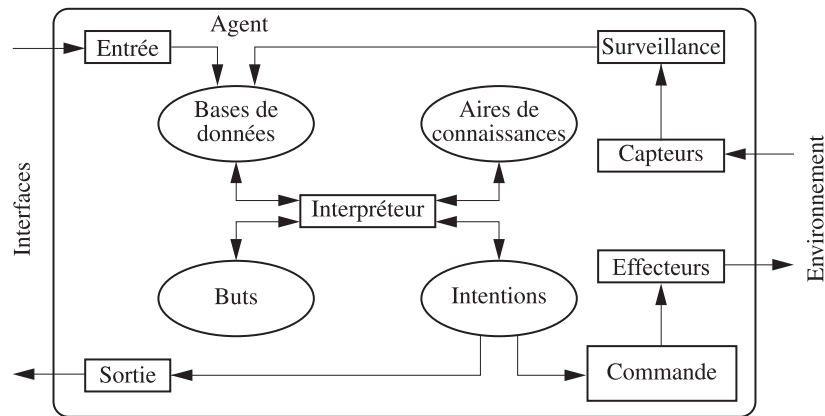


Figure 2.5. Architecture PRS

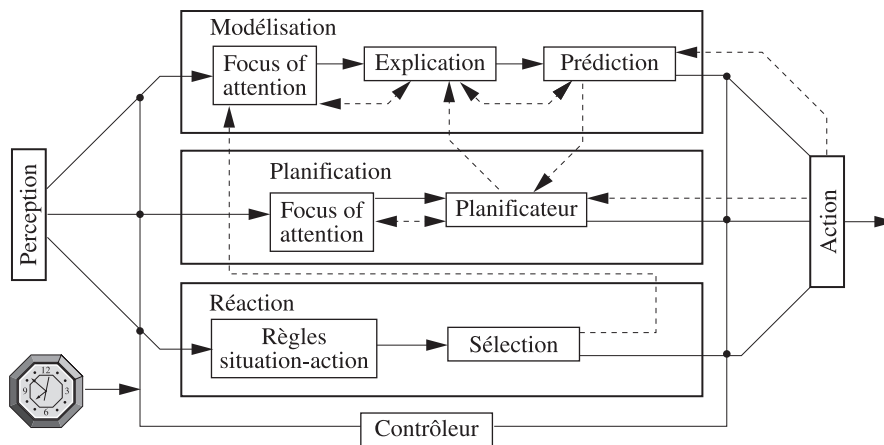


Figure 2.6. Architecture Touring Machine

Cette architecture (voir figure 2.5) est constituée d'une bibliothèque de plans (*Aires de connaissances*), d'une base de croyances représentant des états du monde ou des

déductions internes (*Bases de données*), d'une pile de buts (*Buts*), d'une structure d'intentions (*Intentions*) et d'un ensemble d'opérations manipulant ces informations. Un but est la description du comportement désiré (interne ou externe) de l'agent exprimé en termes de satisfaction, de maintenance ou de vérification de propriétés. Les intentions correspondent à des files de priorité de plans partiellement instanciés, en cours d'exécution ou à exécuter. Une intention peut être active, suspendue ou suspendue sous condition. Un plan correspond à une recette pour satisfaire des buts ou réagir à certaines situations. Un *plan-recette* est constitué : (i) de conditions de déclenchement relatives à des événements, d'activation et de maintenance, portant sur des buts (raisonnement dirigé par les buts) ou sur des croyances (raisonnement dirigé par les données), (ii) d'un corps, structure hiérarchique partielle, décrivant les différentes actions ou sous-buts à satisfaire. Les actions peuvent modifier l'environnement mais elles peuvent aussi manipuler les intentions, croyances et buts de l'agent et conduire à la sélection de plans-recettes (les plans qui les contiennent sont d'un niveau méta).

L'interpréteur reproduit une structure similaire à celle d'IRMA et enchaîne les opérations suivantes : *activation* des plans dont les conditions de déclenchement et d'activation sont satisfaites à partir des événements captés, des croyances et des buts courants ; *choix* d'un ou de plusieurs plans et *ajout* de ceux-ci dans les *Intentions* ; *exécution* d'une intention et *mise à jour* de la file d'événements à partir des perceptions, des croyances et des buts et *retrait* des intentions terminées. La sélection tient compte de critères internes et de critères élaborés par les plans-recettes de métaniveau. Le cycle *uniforme*¹¹ de cette architecture permet de mettre en œuvre un comportement dirigé par les buts tout en satisfaisant des contraintes temps réel et en étant ouverte sur l'environnement.

2.4.3. Architecture d'agent hybride

Le développement d'architectures d'agents autonomes hybrides a débuté en robotique avec notamment *AuRA* [ARK 90] qui utilise un système réactif pour le contrôle de bas niveau du robot et une planification pour la prise de décisions. Les architectures RAP [FIR 89] ou 3T [BON 96] en sont d'autres exemples.

Touring Machine

La *Touring Machine* [FER 92, FER 95] que nous présentons ici vise à implanter des agents réalisant des tâches de navigation contrainte dans des environnements dynamiques. Son architecture (voir figure 2.6) organise les traitements selon une

11. Un cycle uniforme enchaîne de manière cyclique un ensemble d'opérations.

abstraction temporelle et spatiale croissantes, en trois couches horizontales s'intercalant entre la *perception* et l'*action* : *réaction*, *planification* et *modélisation*.

Chaque couche construit sa propre représentation de l'état de l'environnement à partir des informations fournies par la perception et envoie des commandes au système d'action. La couche *réaction* met en œuvre un comportement réactif : activation d'actions précablées en réponse à des stimuli spécifiques de l'environnement. La couche *planification* génère et exécute dynamiquement des plans partiels et hiérarchiques pour l'agent en fonction des tâches et des buts sélectionnés par la couche de modélisation : un *planificateur* linéarise des plans hiérarchiques choisis dans une bibliothèque à partir d'une représentation topologique de l'environnement et d'une représentation filtrée (mécanisme *focus of attention*) de l'environnement perçu. Ce filtrage est contraint par le plan en cours d'élaboration. La couche *modélisation* est dédiée à la coordination des buts internes et des buts externes. Elle représente le comportement des autres entités agissant dans l'environnement, incluant l'agent lui-même. Cette modélisation représente la situation de l'agent, ses croyances, ses désirs, ses intentions. Ces informations sont issues d'un raisonnement abductif à partir des perceptions : filtrage des informations pertinentes (*focus of attention*), générations d'explications sur le comportement des entités (*explication*) et détection de défaut entre le comportement attendu et le comportement réel, résolution des conflits (*prédiction*).

Les couches s'exécutent en parallèle, de manière asynchrone et communiquent entre elles par envois de messages. Chacune des couches ayant accès au système d'action, un *contrôleur* central filtre (censure) les informations provenant *du système de perception* vers les différentes couches, et régule (supprime) les commandes partant des couches vers le *système d'action*. Le comportement de ce contrôleur est spécifié par des règles de contrôle (règles de censure et règles de suppression) spécifiques au domaine d'application. A chaque pas de temps, le contrôleur intervient sur des points de synchronisation situés aux entrées et aux sorties des couches et exécute toutes ses règles de censure et de suppression, actives à cet instant. Ce contrôle global nécessite un effort de conception important afin d'analyser, prévoir et interdire les possibles interactions entre les couches. Notamment, le concepteur est supposé avoir écrit les règles de suppression de telle sorte qu'elles soient cohérentes (une seule règle active à un instant donné) et qu'elles couvrent l'ensemble des situations que l'agent pourra rencontrer. La cohérence de ces règles de contrôle est un problème important. Bien que la couche modélisation puisse modéliser les autres agents de l'environnement, aucun moyen de communication explicite avec les autres agents n'est permis dans cette architecture.

2.5. Architectures d'agents interagissants

L'évolution naturelle des architectures précédentes introduit des capacités d'interaction avec les autres agents du système. Le passage d'une architecture d'agent autonome à une architecture d'agent interagissant fait apparaître des problèmes de conception spécifiques. En effet, au-delà de la définition des interactions proprement dites (voir chapitre 3), les influences des interactions sur les décisions de l'agent et inversement, doivent être modélisées et traitées.

2.5.1. Architecture d'agent réactif

Peu d'architectures d'agents réactifs appartiennent à la catégorie des architectures d'agents interagissants, dans la mesure où les seules interactions mises en œuvre dans de tels systèmes le sont, généralement, *via* la perception des actions exécutées dans l'environnement.

Une architecture échappant à cette tendance est l'architecture PACO (PAtterns de COordinations) [DEM 93]. Les interactions avec les autres agents et avec l'environnement se font par un ensemble de forces dont la combinaison, selon un schéma défini par le concepteur, détermine le déplacement des agents dans l'environnement. Une solution partielle correspond à un agent positionné de manière stable dans l'environnement, une solution globale étant trouvée dès que les agents sont tous stabilisés. Cette architecture a donné lieu à diverses applications en analyse d'images et en robotique [DEM 91b, HAS 92].

L'architecture d'un agent PACO représente l'état de l'agent (*masse* relative à l'importance de l'agent dans la solution globale, *positions*, *vitesse*, *accélération*), les informations liées à son contrôle selon trois champs (perception, communication, action). Le *champ de perception* fixe le sous-ensemble de l'environnement que l'agent peut percevoir. Le *champ de communication*, quant à lui, fixe le sous-ensemble d'agents avec lesquels l'agent peut interagir. Le *champ d'action*, enfin, détermine le sous-ensemble de l'environnement dans lequel l'agent peut se déplacer. Le cycle de contrôle enchaîne trois phases : (i) réglage des champs de perception et de communication, (ii) calcul des forces avec l'environnement, avec les autres agents, et bilan des forces exercées sur l'agent, (iii) réglage du champ d'action en fonction de l'état de l'agent, détermination de l'action de déplacement et calcul d'une nouvelle position de l'agent. Comme nous le voyons, au lieu de sélectionner une action en réponse à un stimulus comme nous pouvions l'avoir dans MANTA, cette architecture calcule l'intensité du seul comportement possible (déplacement) en intégrant l'interaction avec les autres agents et la perception de l'environnement.

Tout comme nous pourrions également le remarquer lors de la description du système Eco-Résolution dans la section suivante, l'interaction est banalisée au sein de l'architecture de l'agent. Elle constitue une nouvelle source d'information s'ajoutant à celle issue de la perception, pouvant déclencher des comportements en réaction.

2.5.2. Architectures d'agents délibératifs

De nombreuses architectures d'agents délibératifs appartiennent au groupe des architectures d'agents interagissants. Comme pour les architectures précédentes, l'interaction peut être traitée comme une source d'information supplémentaire et ne pas faire l'objet d'un raisonnement dédié. Avec le développement d'interactions sophistiquées *via* les actes de langage ou les protocoles d'interaction, certaines architectures ont développé des modules dédiés à la gestion des interactions. C'est ce que nous verrons au travers des systèmes COSY, ADEPT, etc.

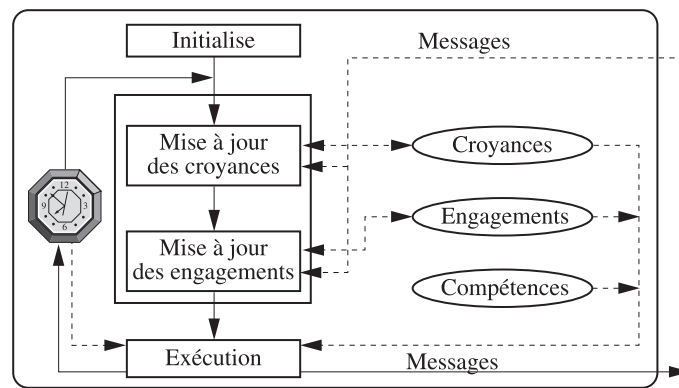


Figure 2.7. Architecture AOP

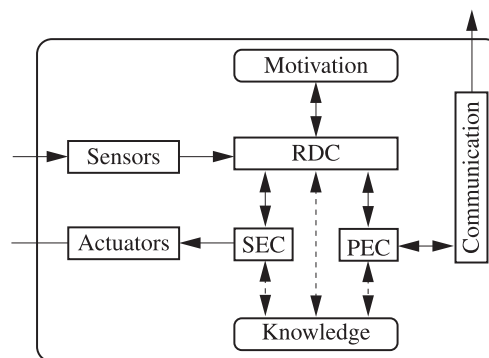


Figure 2.8. Architecture COSY

2.5.2.1. Agent Oriented Programming

Le paradigme *Agent Oriented Programming* (AOP) [SHO 93] propose un système constitué d'un langage formel pour décrire les états mentaux d'un agent, d'un langage de programmation interprété dont la sémantique correspond à celle des états mentaux, d'un « agentifieur » traduisant un logiciel en un agent fonctionnant avec un ensemble de capteurs et d'effecteurs réels, du même ordre que ce qui a été proposé pour les automates situés [ROS 86]. L'architecture d'agent proposée se distingue des architectures BDI que nous avons pu décrire ci-dessus par les primitives représentées au sein de l'*état mental*. Ce sont les croyances, les compétences, les choix et les engagements. La première réalisation du langage correspondant (*Agent0*) permet de représenter des faits, des actions conditionnelles ou inconditionnelles (*Do*, *Refrain*), internes ou de communication, et des règles d'engagement qui décrivent la manière dont l'agent s'engagera en fonction de son état mental et des messages reçus. Dans ce langage, trois types de messages ont seulement été définis pour mettre à jour les croyances de l'agent (*Inform*), demander (*Request*) ou annuler une demande (*Unrequest*) en mettant à jour les engagements. Le langage de programmation est constitué d'un interpréteur (voir figure 2.7) qui met en œuvre un cycle de contrôle constitué de deux étapes : mise à jour des états mentaux à partir des messages reçus, génération et exécution des engagements courants définis à partir des règles d'engagements.

Cette première version très simple et rudimentaire du langage a été étendue dans PLACA (*PLAnning Communicating Agents*) [THO 95]. Les ajouts ont concerné les communications et le raisonnement sur les buts, la planification. La modalité d'*Intention* a été introduite et le cycle de l'interpréteur a été modifié en y introduisant des étapes, dépendantes du temps, de construction et de raffinement de plans. Une architecture similaire à AOP sert d'élément de base dans la plate-forme KAOS [BRA 96] avec l'adjonction d'un module explicite de gestion des conversations. Cette architecture est ainsi plus proche de l'architecture décrite ci-dessous.

2.5.2.2. COSY

Le développement de l'architecture COSY [BUR 92, HAD 95] s'est situé dans le contexte de l'étude de la coopération entre systèmes à base de connaissances pour la gestion de trafic routier, pour des applications dans le domaine manufacturier également. A la différence de l'architecture précédente, la gestion des interactions fait l'objet d'un module dédié. Les mécanismes et représentations mis en œuvre sont beaucoup plus élaborés afin de gérer l'exécution de protocoles d'interaction. Cinq modules la composent (voir figure 2.8) : *actuators*, *sensors*, *communication*, *cognition*, *motivation*. Le module *sensors* met à jour la représentation de l'état courant de l'environnement au sein de l'agent et transmet au module de *cognition* les événements y survenant ; *actuators* exécute les actions physiques sur l'environnement ; *communication* gère l'envoi et la réception de messages avec les autres agents ;

motivations gère l'ensemble des buts à long terme, des rôles et des préférences de l'agent et *cognition* sélectionne, exécute et supervise les actions (sur l'environnement, de raisonnement et de communication) de l'agent à partir de l'évaluation de la situation courante de l'environnement. Ce module représente dans *Knowledge* une bibliothèque de plans-recettes¹² (BDPR) et de protocoles de coopération¹³ (BDPC). La fonction RDC sélectionne les tâches et les plans à exécuter et lance l'exécution de sous-plans dans la fonction d'exécution de scripts (SEC) ou de protocoles dans la fonction d'exécution de protocole (PEC). La fonction SEC délègue l'exécution des actions au module *actuators*. La fonction PEC exécute les protocoles de coopération en préparant les messages à émettre, en les passant au module *communication*, et traite, en tenant compte des protocoles d'interaction, les messages reçus en provenance du module *communication*. Nous trouvons ainsi dans cette architecture des modules dédiés à l'implantation de la facette E pour l'accès à l'environnement (*actuators* et *sensors*), de la facette A pour le raisonnement interne (SEC, RDC) et de la facette I pour les interactions (*communicator*, PEC). Le contrôle de la coopération est mélangé avec le raisonnement du domaine.

Poursuivant dans la démarche de réalisation des interactions par un module dédié, le système IMAGINE [HAU 94, STE 96] ou ARCHON [WIT 92], que nous ne détaillerons pas ici, issus de projets Européens, proposent une architecture en couches permettant d'intégrer et de faire coopérer des programmes préexistants dans le cadre du travail coopératif. Au-delà d'une couche intégrant les programmes préexistants, et d'une couche chargée du contrôle de l'ensemble et de la coopération avec les autres agents, une couche communication est chargée de l'envoi et de la réception de messages formulés par des actes de langage.

2.5.3. Architectures d'agents hybrides

L'architecture InteRRaP vise à fournir un outil pour construire des systèmes complexes et dynamiques dans des domaines tels que la robotique [MUL 94], l'ordonnancement coopératif [FIS 95a] dans lesquels les agents ont des ressources limitées, interagissent avec les autres agents et sont dirigés par les buts, tout en

12. Les plans-recettes sont implantés sous la forme de scripts constitués de conditions d'activation, des ressources nécessaires, d'un corps exprimé par une structure arborescente dont les nœuds correspondent à des appels à d'autres scripts (plans-recettes ou protocoles) ou à des actions sur l'environnement ou de raisonnement.

13. Les protocoles de coopération sont représentés sous la même forme que les plans-recettes à la différence que les nœuds de la structure arborescente correspondent à des actions primitives d'envoi et de réception de messages ou à l'appel d'autres protocoles (à la différence d'AOP, la formalisation de ces actions primitives [HAD 95] ne s'appuie pas sur la théorie des actes de langage).

combinant un comportement réactif et coopératif. Cette architecture d'agent s'intègre dans un environnement de développement et de test d'applications multi-agents AgenDA [FIS 96]. Elle s'appuie sur la théorie BDI et connaît de nombreuses évolutions actuellement [JUN 99] en termes de formalisation et d'implantation.

Cette architecture [MUL 94, MUL 96] (voir figure 2.9), inspirée de l'architecture autonome réactive RAP [FIR 89], est constituée de trois couches fonctionnant en parallèle : *planification coopérative (PC)*, *planification locale (PL)* et *comportement (C)*. Chacune des couches possède une structure uniforme constituée de deux fonctions générales : *reconnaissance de situation* et *activation de but (SB)*, *planification* et *ordonnancement (PO)*. Une base de connaissance représente hiérarchiquement les informations manipulées par les différentes fonctions sur chacune des couches, tout en assurant la révision de croyances et l'abstraction des informations : croyances sur l'état de l'environnement (*modèle du monde*) sur la couche *C*, croyances que l'agent a sur lui-même (*modèle mental*) sur la couche *PL*, croyances que l'agent peut avoir sur les autres agents du système (*modèle social*) sur la couche *PC*. Un composant constitué des fonctions de perception, de communication et d'action réalise l'interface avec l'environnement et les autres agents. La couche *C* réagit aux évolutions de l'environnement par l'utilisation de schémas de réaction (condition-action) déclenchés par des événements externes, et réalise des activités routinières (procédures de comportement). La couche *PL* contrôle la planification locale d'un agent en sélectionnant dans une bibliothèque de plans-recettes. Ceux-ci peuvent faire appel aux procédures de comportement de la couche *C* ou à d'autres sous-plans. La couche *PC* élargit les capacités de planification à un univers multi-agent en manipulant des plans communs (bibliothèque de plans communs) avec d'autres agents et en utilisant des protocoles de négociation, et des stratégies de négociation. Elle produit des plans partiels communs comportant des points de synchronisation. Elle reconnaît les situations d'interaction, dérive de nouveaux buts en fonction des situations détectées et choisit des protocoles et des stratégies pour la négociation afin que les agents trouvent des accords avec d'autres agents (plans communs). Planification, ordonnancement et exécution se déroulent à deux niveaux différents : niveau méta (traitement des protocoles en fonction des stratégies afin de sélectionner un plan), niveau objet (génération des plans candidats pour la négociation).

Deux mécanismes de contrôle hiérarchique régulent les échanges entre les couches : requêtes ascendantes d'activation, signaux descendants d'engagements pour l'exécution. Une couche i ne devient active que si la couche inférieure ne peut pas traiter la situation courante : le processus PO_i de la couche i , envoie une demande d'activation à SB_{i+1} de la couche supérieure. Le résultat du traitement est renvoyé à PO_i . Sur une couche i , PO_i exécute ses décisions en communiquant ses engagements à la couche inférieure (PO_{i-1}).

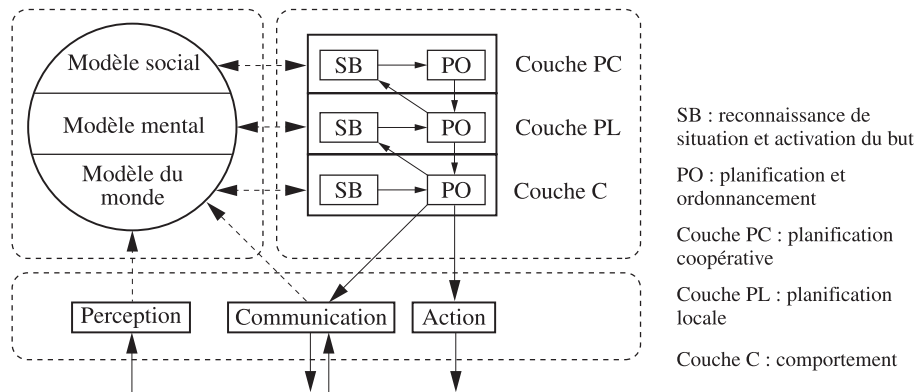


Figure 2.9. Architecture InteRRaP (d'après [FIS 95b])

Par rapport aux architectures similaires telles que la Touring Machine, elle ajoute un composant supplémentaire dédié à la gestion de la coopération avec les autres agents, mais pas de module dédié à la gestion des interactions avec les autres agents comme nous pouvions en avoir un dans les architectures d'agents délibératifs précédentes.

2.6. Architectures d'agents sociaux

Le développement d'architectures d'agents sociaux est une tendance naturelle des recherches en SMA du fait des applications de plus en plus complexes qui sont abordées et des travaux réalisés actuellement sur les organisations. Le passage d'une architecture d'agent autonome ou interagissant à une architecture d'agent social fait apparaître des problèmes de conception spécifiques. Au-delà de la gestion des données liées à la facette *O* (statuts, dépendances, relations entre les agents), l'architecture doit modéliser et traiter les influences de ces dernières sur les décisions et les interactions de l'agent et inversement.

2.6.1. Architectures d'agents réactifs

Nous avons choisi de présenter deux architectures d'agents réactifs qui toutes deux manipulent une organisation définie par le concepteur. Dans le cas de la première architecture, elle ne peut pas être adaptée. Au sein de la deuxième architecture, il est possible de la modifier tout en l'utilisant pour contraindre la solution.

2.6.1.1. *Eco-Résolution*

Eco-Résolution [FER 89, FER 91] propose de remplacer les méthodes classiques de résolution de problèmes par une approche multi-agent dans laquelle la solution correspond à l'obtention de configurations stables d'une population d'agents au travers de leurs interactions. Cette approche a été appliquée au taquin [DRO 92], à PENG1, etc. L'architecture de l'agent permet de représenter un *but*, des *dépendances*, un *état interne*. Les deux premiers éléments représentent explicitement des relations statiques que l'agent a avec d'autres agents. En effet, le but correspond à un autre agent avec lequel l'agent doit être dans une relation de satisfaction, les dépendances sont les agents dont l'agent est lui-même le but. Ces relations induisent un ordre sur la résolution dans la mesure où ces agents ne pourront être satisfaits qu'après lui. Chaque agent cherche à atteindre un état interne stable, satisfaisant au mieux son but. L'évolution de l'état interne est régi par un automate à états finis dont les transitions vont conduire l'agent à mettre en œuvre des comportements de satisfaction de son état, d'agression (déclenchement d'une fuite chez un autre agent) et de fuite (changement de position dans l'environnement). L'interaction est intégrée directement dans le code des comportements de base de l'agent *via* des messages de demande de fuite et d'échange d'états de satisfaction.

2.6.1.2. *PACORG*

Le système PACORG [BAE 98] a étendu le modèle de SMARRPS [FER 97] fondé sur PACO afin d'explicitier la gestion de l'organisation et l'a appliqué à la généralisation cartographique [BAE 96]. Le *champ de communication* défini dans PACO a été subdivisé en : *champ de classe* (agents appartenant à la même classe que l'agent), *champ d'objet* (agents appartenant au même objet géographique) et *champs de groupes*. Une relation de subordination est également prise en compte lors de la génération de l'action à exécuter. Le cycle initial de PACO (voir paragraphe 2.5.1.1) a également été modifié pour faire intervenir la structure organisationnelle. Ce sont la phase (i) consistant au réglage des champs, et la dernière phase (iii) lors de la génération du déplacement. La modification de la phase (i) introduit ainsi la possibilité de faire évoluer l'organisation définie. La modification de la phase (iii) introduit l'influence de l'organisation sur la génération du déplacement. Tout comme nous l'avons remarqué dans PACO, la mise en place de la gestion de l'organisation est totalement intégrée dans le cycle global de l'architecture au sein de la fonction de calcul des forces.

2.6.2. *Architectures d'agents délibératifs*

De plus en plus d'architectures d'agents délibératifs permettent de manipuler ou de prendre en compte les organisations. Certaines architectures implémentent les fonctionnalités liées à l'organisation, uniquement au travers de la représentation

d'une organisation qui est prise en compte dans le traitement. C'est le cas par exemple de l'architecture ADEPT.

2.6.2.1. ADEPT

Le système ADEPT [JEN 00a, JEN 00b] se situe dans la lignée des systèmes GRATE [JEN 95] et ARCHON [JEN 96, WIT 92], architectures d'intégration d'applications existantes dans un SMA. Il a été appliqué à la gestion de processus administratifs. L'architecture proposée appartient à la famille des architectures BDI.

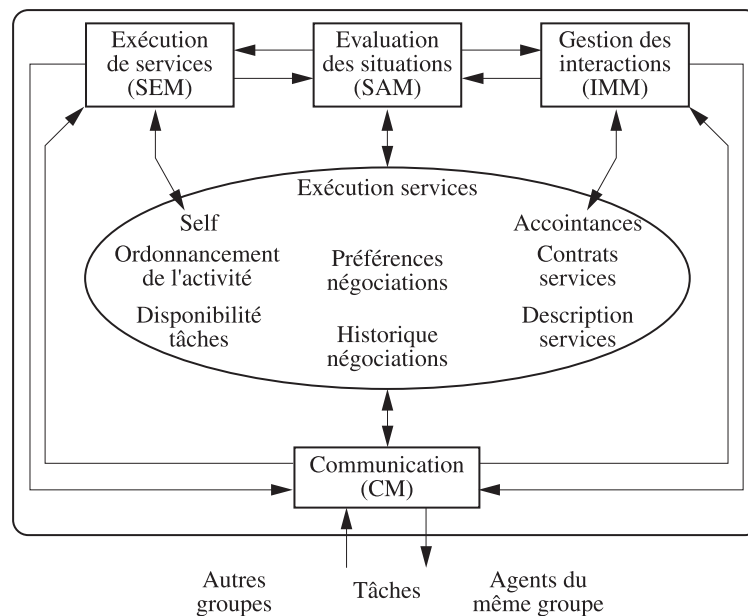


Figure 2.10. Architecture de la couche coopération/contrôle d'ADEPT

Des modalités supplémentaires ont été définies pour représenter des activités conjointes entre agents (*joint intentions*) [JEN 93b]. L'organisation apparaît dans cette architecture *via* la notion de groupes (*agency*) auxquels peuvent appartenir les agents et dont ils peuvent être un représentant vis-à-vis des agents externes au groupe.

Chaque agent intègre un ensemble de *services* préexistant qu'il peut activer dans le cadre de l'exécution de tâches administratives locales ou globales par coopération avec les autres agents. L'architecture sépare nettement ces deux aspects : une couche

domaine gère les services intégrés, une couche *coopération/contrôle* contrôle l'activation de la couche inférieure et assure la coordination de l'agent avec les autres. Au sein de la couche *coopération/contrôle* (voir figure 2.10) sont représentés : (i) les capacités des autres agents ainsi que la structure des groupes (*accointances*), les services accessibles dans la couche *domaine* de l'agent (*self*), (ii) l'état courant des tâches locales et externes. L'ensemble *self* contient également une bibliothèque de plans-recettes, les intentions et les buts de l'agent, ainsi que les intentions communes. L'ensemble de ces informations est accessible par trois modules : *exécution des services* (SEM), *évaluation des situations* (SAM), et *gestion des interactions* (IMM). Un module de *communication* assure la communication par l'envoi de messages avec les autres agents. Le SAM a un fonctionnement similaire à celui de l'interpréteur de PRS, à la différence qu'il peut, en plus, recevoir des événements issus de messages provenant d'autres agents et qu'il gère des plans pouvant nécessiter la collaboration avec d'autres agents. Il régule l'activité de l'agent en privilégiant activité locale ou externe selon les accords de service en cours ou en prévision. Il maintient ainsi un état des différents accords en cours et prend en compte les exceptions liées à l'exécution des services. A partir des tâches à exécuter localement que lui fournit le SAM, le SEM est responsable de leur planification, de leur exécution dans la couche *domaine* et de leur supervision. De manière similaire, le SAM peut déclencher le module IMM afin de gérer la négociation de services avec les autres agents par argumentation [SIE 97]. Au sein du module IMM, une fonction d'*évaluation* accepte ou non les propositions provenant d'autres agents, une fonction de *raisonnement stratégique* décide de la manière de répondre (coopérative, compétitive par exemple) en fonction de l'appartenance des agents aux groupes définis et une fonction de *raisonnement tactique* remplit la contre proposition. Lors de l'envoi de messages, le IMM active le module de *communication*. Cette architecture prend en compte les groupes définis. Par contre, aucun mécanisme n'est défini pour les faire évoluer dynamiquement.

2.6.2.2. Cooperative Information Agent

L'architecture CIA [VER 98] (*Cooperative Information Agent*) (voir figure 2.11) est similaire à l'architecture précédente, en y ajoutant la gestion de contrats. Elle a été utilisée pour l'accès coopératif à différentes sources d'informations. L'organisation est définie d'une part, par des *contrats* décrivant les comportements de communication qui sont autorisés ou obligatoires entre des fournisseurs de services et d'autre part, par trois types de relation : pouvoir (relation institutionnelle fixée permettant à un agent de commander l'exécution d'un service), autorité (qui doit être explicitement accordée et qui a une durée limitée), charité (un agent peut demander l'exécution d'un service sans garantie que le service soit réalisé).

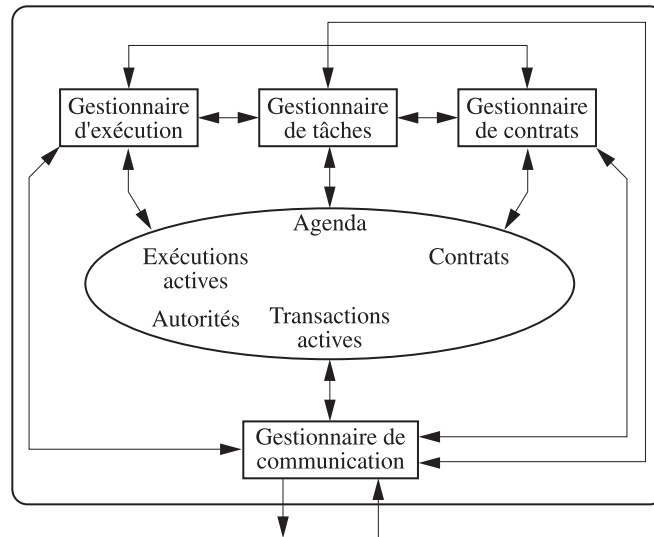


Figure 2.11. Architecture CIA (d'après [VER 98])

Dans cette architecture, quatre modules sont responsables chacun d'une des activités principales de l'agent : gestionnaire de tâches, gestionnaire de contrats, gestionnaire des communications et interactions, gestionnaire de l'exécution des services. Le *gestionnaire de contrats* est responsable de la négociation de nouveaux contrats et du suivi des contrats établis (définition des étapes à suivre lorsqu'un contrat est rompu). Le *gestionnaire de tâches* planifie, ordonnance et suit l'exécution des tâches que l'agent doit exécuter. Les tâches sont constituées d'actions locales ou d'actions nécessitant des communications (*transactions*). Les éléments de base des transactions sont les actes de langage [DIG 95]. Les transactions sont utilisées dans les contrats décrivant les engagements entre deux agents. Le *gestionnaire de communication* initie et gère les échanges de messages entre les agents. Le *gestionnaire d'exécution* gère les actions locales et les services qu'un agent peut réaliser pour d'autres agents (vérification que les autres agents peuvent demander ce service, disponibilité des services demandés ainsi que des services pouvant être offerts par d'autres agents) à l'aide des relations définies dans l'organisation. Au cas où aucune relation n'existe, le *gestionnaire de contrat* est activé pour mettre en place un contrat pour l'exécution du service. A partir des obligations que lui communique le *gestionnaire de contrat*, le *gestionnaire de tâches* maintient un agenda des actions à exécuter. Les obligations peuvent être issues des tâches de l'agent mais aussi des contrats réalisés.

2.6.2.3. Autres architectures

D'autres architectures rejoignent les exemples précédents quant au degré de développement des mécanismes liés à l'organisation, même si les organisations gérées sont d'une autre nature. C'est, par exemple, le cas de l'architecture ASIC [BOI 93, BOI 97] (*Architecture for Social and Individual Control*) où l'organisation est exprimée en termes de relations et de rôles entre agents. Elle enchaîne les opérations liées au contrôle individuel implémentant le contrôle local de l'agent selon une hiérarchie de contrôle, et celles liées au contrôle social implémentant le contrôle global du système. Les opérations liées au contrôle social sont regroupées en un module pour la gestion des interactions entre les agents et en un module organisation permettant de prendre en compte les structures organisationnelles définies et de filtrer les informations reçues ou émises par l'agent.

D'autres architectures vont plus loin dans la réalisation des fonctionnalités liées à l'organisation, en y ajoutant des mécanismes pour la mise en place de l'organisation elle-même et sa prise en compte dans les traitements. C'est le cas, par exemple, dans [LES 98], où est proposée une architecture dans laquelle une couche dédiée à la mise en place d'une organisation en termes de responsabilités de tâche vient s'ajouter à une couche dédiée à la coordination dans laquelle le mécanisme GPGP basé sur le langage de tâches TAEMS [DEC 95]¹⁴ est mis en œuvre. L'architecture d'agent définie dans SAM [BOU 92] est fondée sur les notions d'actes de langage et d'engagement. L'organisation y est vue comme l'ensemble des engagements tenus par les agents du système. L'architecture est composée de quatre modules : perception, communication, organisation et décision. Ce dernier module joue un rôle central dans l'architecture en orchestrant le fonctionnement des autres modules. Le module de perception construit une représentation de la situation perçue. Le module de communication construit une interprétation des actes de langage reçus par l'agent. La gestion proprement dite des protocoles d'interaction relève du module de décision. Le module d'organisation définit les engagements tenus par l'agent en fonction de son rôle. Le module de décision, enfin, détermine les actions physiques et les communications qui doivent être exécutées.

D'autres architectures manipulent les organisations au travers des réseaux de dépendances. C'est le cas par exemple de l'architecture STARS [ALL 98] utilisée pour des applications de supervision, où un ensemble de modules dédiés à la gestion de dépendances temporelles entre agents sont définis [ALL 00]. Une autre architecture de ce type est celle de DIMA [GUE 96] dans laquelle une couche supérieure est dédiée à un module de supervision qui par l'intermédiaire de l'interprétation d'ATN contrôle l'activité des modules opérant dans la couche inférieure : perception, raisonnement, action. Le module de perception met à jour un

14. Se reporter au chapitre 14 : « Modèles de coordination d'agents cognitifs ».

ensemble de données sur l'état de l'environnement que peut consulter le module de supervision ou le module de raisonnement. Le module d'action est chargé de la communication avec les autres agents (réception et émission de messages) et de l'action sur l'environnement. Le module de raisonnement définit la réponse adaptée aux informations en provenance des modules de supervision et d'action, à partir de ses propres connaissances. Le raisonnement mis en place en son sein prend en compte des réseaux de dépendances par rapport aux ressources, mis à jour tout au long de l'activité de l'agent. A notre connaissance, il n'y a pas, dans le domaine, d'architecture d'agent sociale hybride.

Raisonnement	Agent réactif	Agent hybride	Agent délibératif
Coordination			
Agent autonome	Subsumption PENGI, MANTA	Touring Machine	<i>IRMA, PRS/dMARS</i>
Agent interagissant	PACO, SMARRPS	<i>InteRRap</i>	IMAGINE, ARCHON, AOP, COSY, GRATE*
Agent social	ECO, PACORG		ADEPT, ASIC, STARS, SAM, DIMA, CIA.

Tableau 2.1. Caractérisation des architectures d'agents selon les visions coordination et raisonnement. En italique sont indiquées les architectures de type BDI

2.7. Architectures

Nous rassemblons dans le tableau 2.1 les différentes architectures que nous avons décrites ou citées en les classant selon les dimensions de *coordination* et de *raisonnement* présentées dans la section 2.3. Les différentes présentations montrent que les motivations donnant lieu au développement des architectures décrites sont multiples : outil d'intégration de logiciels existants dans un SMA (ARCHON, ADEPT, IMAGINE), outil de développement pour des applications précises (PRS, Subsumption, SMARRPS, ECO, COSY) ou outils de simulation (PENGI, MANTA). Au-delà de ces propriétés extrinsèques, nous pouvons nous intéresser aux caractéristiques internes des architectures en retenant dans un premier temps la *décomposition* des modules au sein de chacune des architectures, puis les *interactions* entre ces modules (flux de données et de contrôle) qui régissent le fonctionnement de l'architecture.

2.7.1. Décomposition modulaire

Une démarche fonctionnelle peut conduire à différentes décompositions [FOD 83]. Dans le cadre des instances d'architectures présentées, ces décompositions apparaissent par rapport aux fonctions de perception et d'action. Pour certaines architectures, un ou plusieurs modules réalisent chacun une activité complexe ayant accès aux données perçues et aux effecteurs. Cette *décomposition horizontale* (voir figure 2.12) est visible dans l'architecture en couches de la Touring Machine, de l'*architecture subsumption* ou dans les architectures monocouches¹⁵ de ECO, PACO, MANTA. D'autres architectures sont constituées de modules qui participent à différentes activités sans avoir accès *a priori* directement aux données perçues et aux effecteurs. Nous appelons *décomposition verticale modulaire* (voir figure 2.12), le cas où deux modules différents dans un cycle de contrôle unifié ont accès respectivement à la perception et à l'action. C'est ce que l'on retrouve par exemple dans les architectures délibératives de PRS, Agent0, COSY. Nous appelons *décomposition verticale en couches* (voir figure 2.12), le cas où un module réalise l'interface avec le système de perception et d'action. Ce type de décomposition est visible dans l'architecture d'agent hybride d'InteRRap ou dans les architectures d'agents délibératifs pour l'intégration telles que Adept, Archon, Imagine.

2.7.2. Interactions intermodules, cycle de contrôle

Sur cette décomposition modulaire, le contrôle du système va structurer le fonctionnement interne de chacun des modules et entre les modules. Les architectures verticales modulaires et les architectures monocouches ont un seul cycle de contrôle uniforme qui enchaîne les traitements de chacun des modules au sein de l'architecture.

Dans les architectures multicouches (voir figure 2.12), selon la couche considérée, les modules ont des représentations et des algorithmes différents pour exécuter la réaction et la délibération (cas des architectures d'agents hybrides¹⁶) ou les différents comportements (cas des architectures d'agents réactifs). Différentes structures de contrôle définissent la coopération entre ces couches. Ainsi, l'*architecture subsumption* institue une hiérarchie d'*inhibition/activation* dans laquelle un module d'une couche donnée peut inhiber un module d'une couche inférieure. Dans la Touring Machine, un contrôleur central régit les interactions

15. Une telle décomposition pour les architectures d'agents réactifs n'est pas étonnante dans la mesure où ces architectures s'opposaient à la conception traditionnelle d'un découpage fonctionnel dédiant des modules à la perception, à l'action, à la planification, à la décision, etc.

16. L'hypothèse est que délibération et réaction sont sur des échelles temporelles différentes.

entre couches selon un mode de compétition. Dans le cas d’InteRRap, en revanche, la hiérarchie mise en place induit un flux de contrôle inverse au flux de données qui peut être qualifié de méta. Ces architectures en couches pour la plupart mettent en œuvre un cycle uniforme au sein de chacune des couches pour définir l’enchaînement des traitements.

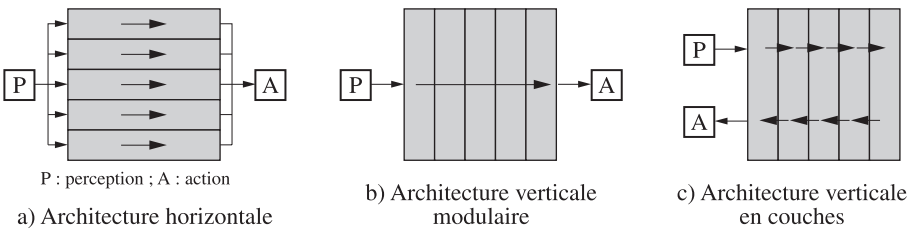


Figure 2.12. *Décomposition modulaire des architectures à partir de [MUL 95]*

Décomposition modulaire	Architectures
Horizontale avec plusieurs couches	Subsumption, Touring Machine,
Horizontale avec une seule couche	Pengi, Manta, ECO-Résolution, SMARPS, PACORG, PACO
Verticale modulaire	IRMA, PRS/dMARS, AOP, COSY,
Verticale en couches	IMAGINE, InteRRap, CIA, ADEPT

Tableau 2.2. *Caractérisation des architectures d’agents selon leur décomposition modulaire*

2.8. Discussion

Les architectures d’agents sont des modèles conceptuels ou implémentés intégrant des matériels ou logiciels divers. Comme nous avons pu le voir, elles sont très nombreuses et diverses. Selon les applications, ces architectures peuvent générer et exécuter des plans d’actions satisfaisant leurs buts, tout en réagissant à des événements inattendus provenant de l’environnement, gérer les changements et les incertitudes, interagir avec les autres agents avec des langages sophistiqués et enfin, être capables de participer à des organisations élaborées avec les autres agents. Le tour d’horizon que nous venons d’en faire a permis de les classer selon leurs capacités de raisonnement et de coopération dans un univers multi-agent d’une part, et selon leurs propriétés architecturales d’autre part.

Ce domaine de recherche est en évolution continue. Ainsi, au fur et à mesure des progrès effectués et des contraintes applicatives, de nouvelles architectures apparaissent. Ainsi, des architectures d'agents *récurives*, c'est-à-dire spécifiées, conçues et implantées comme étant elles-mêmes des systèmes multi-agents [FIS 99, MAR 97, OCC 97] commencent à faire leur apparition. Les capacités de raisonnement sont également enrichies avec des capacités de raisonnement temporel [BOU 97, VER 90] ou [ALL 98], la gestion d'émotions [AUB 96, BAT 94], etc.

Un travail important reste à faire également afin de développer des théories, des architectures et des langages pour la spécification, l'implémentation et la validation d'agents. Concernant l'évaluation et la comparaison de ces architectures, il faut pouvoir fournir des critères permettant à partir d'une caractérisation du domaine d'application, de choisir l'architecture adéquate [MUL 99]. Une autre perspective consiste à tirer profit des efforts réalisés au sein de la communauté du génie logiciel qui œuvre dans le sens de fournir des langages de description d'architecture [MON 97], de récolter des styles d'architecture et de s'unir à leurs efforts afin par exemple de constituer des patrons de conceptions d'architectures d'agents [KEN 97], de concevoir des architectures plus modulaires [GUI 99].

2.9. Bibliographie

- [AGR 87] AGRE P.E., CHAPMAN D., « Pengi: an implementation of a Theory of Activity », dans *Proceedings of AAIL*, Morgan Kaufman, p. 268-272, 1987.
- [ALL 90] ALLEN J., HENDLER J., TATE A. (dir.), *Readings in Planning*, Morgan Kaufmann, San Mateo, Californie, août 1990.
- [ALL 98] ALLOUCHE M.K., Une société d'agents temporels pour la supervision de systèmes industriels, Thèse de doctorat, Ecole nationale supérieure des mines de Saint-Etienne, Université Jean Monnet, octobre 1998.
- [ALL 00] ALLOUCHE M.K., BOISSIER O., SAYETTAT C., « Temporal Social Reasoning in Dynamic Multi-Agent Systems », *4th International Conference on Multi-Agent Systems*, p. 23-30, 2000.
- [ARK 90] ARKIN R.C., « Integrating behavioral, perceptual, and world knowledge in reactive navigation », dans [MAES 90a], p. 105-122, 1990.
- [AUB 96] AUBÉ M., SENTENI A., *Emotions as Commitments Operators: A Foundation for Control Structure in Multi-Agents Systems*, Springer-Verlag, LNAI n° 1038, p. 13-25, 1996.
- [BAE 96] BAEIJIS C., DEMAZEAU Y., ALVARES L., « SIGMA : application of multi-agent systems to cartographic generalization », dans Van de Velde W., Perram J. (dir.), *Agents Breaking away, 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Springer-Verlag, LNAI n° 1038, p. 163-176, 1996.

- [BAE 98] BAEIJIS C., Fonctionnalité émergente dans une société d'agents autonomes. Etude des aspects organisationnels dans les Multi-Agents réactifs, Thèse de doctorat, INP Grenoble, 1998.
- [BAT 94] BATES J., BRYAN LOYALL A., SCOTT REILLY W., « An architecture for action, emotion and social behavior », dans C. Castelfranchi et E. Wenner (dir.), *Artificial Social Systems (Proceedings of the 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-92))*, Elsevier North-Holland, p. 55-68, 1994.
- [BOI 93] BOISSIER O., Problème du contrôle dans un système intégré de vision, utilisation d'une approche multi-agent, Thèse de Doctorat, INP Grenoble, janvier 1993.
- [BOI 97] BOISSIER O., DEMAZEY Y., « Une architecture multi-agent pour des systèmes de vision ouverts et décentralisés », *Technique et science informatiques*, vol. 16, n° 8, p. 1039-1062, 1997.
- [BON 96] BONASSO R.P., KORTENKAMP D., MILLER D.P., SLACK M., « Experiences with an architecture for intelligent, reactive agents », dans Wooldridge M.J., Müller J.P., Tambe M. (dir.), *Intelligent Agents: Agent Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1037, p. 187-202, 1996.
- [BOU 92] BOURON T., Structures de communication et d'organisation pour la coopération dans un univers multi-agent, Thèse de doctorat, Paris VI, 1992.
- [BOU 97] BOURY-BRISSET A.-C., MOULIN B., « Un modèle d'agent intégrant des mécanismes de raisonnement multiples », *Revue d'Intelligence Artificielle*, vol. 11, n° 1, p. 73-107, 1997.
- [BRA 96] BRADSHAW J.M., DUTFIELD S., BENOIT P., WOOLLEY J.D., « KaoS: Toward an industrial-strength generic agent architecture », dans Bradshaw J.M. (dir.), *Software Agents*, AAAI/MIT Press, p. 375-418, 1996.
- [BRA 87] BRATMAN M.E., *Intentions, Plans, and Practical Reason*, Harvard University Press, 1987.
- [BRA 88] BRATMAN M.E., ISRAEL D.J., POLLACK M.E., « Plans and resource-bounded practical reasoning », *Computational Intelligence*, n° 4, p. 349-355, 1988.
- [BRO 86] BROOKS R., « A robust layered control system for a mobile robot », *IEEE journal of robotics and automation*, vol. 2, n° 1, p. 14-23, 1986.
- [BRO 91] BROOKS R., « Intelligence without representation », *Artificial Intelligence*, n° 47, p. 139-159, 1991.
- [BUR 92] BURMEISTER B., SUNDERMEYER K., « Cooperative problem solving guided by intention and perception », dans Werner E., Demazeau Y. (dir.), *Decentralized AI III - Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, Elsevier North-Holland, p. 77-92, 1992.
- [CAS 92] CASTELFRANCHI C., MICELI M., CESTA A., « Dependence relations among autonomous agents », dans Werner E., Demazeau Y. (dir.), *Decentralized AI III - Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, Elsevier North-Holland, p. 215-230, 1992.

- [CAS 95] CASTELFRANCHI C., « Guarantees for Autonomy in Cognitive Agent Architecture », dans Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents I: Agent Theories, Architectures and Languages*, Springer-Verlag, p. 56-70, 1995.
- [CAS 98a] CASTELFRANCHI C., « Modelling Social Action for AI Agents », *Artificial Intelligence*, vol. 103, n° 1-2, p. 157-182, 1998.
- [CAS 98b] CASTELFRANCHI C., « Towards an Agent Ontology: Autonomy, Delegation, Adaptivity », dans d'Aloisi D. (dir.), *Italian journal of the AI association*, numéro spécial Agents, 1998.
- [CAV 98] CAVEDON L., SONENBERG L., « On social commitment, roles and preferred goals », *Proceedings of the 3rd International Conference on Multi-Agent Systems*, p. 80-87, 1998.
- [CHA 90] CHAPMAN D., Vision Instruction and Action, PhD Thesis, MIT, 1990.
- [COH 90] COHEN P.R., LEVESQUE H.J., « Intention is choice with commitment », *Artificial Intelligence*, n° 42, p. 213-261, 1990.
- [DEC 95] DECKER K.S., Environment centered analysis and design of coordination mechanisms, PhD Thesis, Univ. Of Massachusetts at Amherst, Computer Science Dpt, 1995.
- [DEM 91a] DEMAZEAU Y., MÜLLER J.P., « From reactive to intentional agents », dans Demazeau Y., Müller J.P. (dir.), *Decentralized Artificial Intelligence II, Proceedings of the 2nd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-90)*, Elsevier North-Holland, p. 3-10, 1991.
- [DEM 91b] DEMAZEAU Y., « Coordination patterns in multi-agent worlds: application to robotics and computer vision », dans *proceedings of IEE colloquium on intelligent agents*, IEEE, Londres, 1991.
- [DEM 93] DEMAZEAU Y., « La plate-forme PACO et ses applications », dans *2^e journée nationale du PRC-IA sur les Systèmes Multi-Agents*, PRC-IA, Montpellier, 1993.
- [DEM 95] DEMAZEAU Y., « From interactions to collective behaviour in agent-based systems », dans *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo, p. 117-132, 1995.
- [DEN 87] DENNET D.C., *The intentional Stance*, MIT Press, 1987.
- [DIG 95] DIGNUM F., WEIGAND H., « Communication and Deontic Logic », dans R. Wieringa, R. Feenstra (dir.), *Information systems, Correctness and Reusability*, p. 242-260, 1995.
- [DRO 92] DROGOUL A., DUBREUIL C., « Eco Problem Solving: Results of the N-Puzzle », dans WERNER E., DEMAZEAU Y. (dir.), *Decentralized Artificial Intelligence III, Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, Elsevier North-Holland, p.283-296, 1992.
- [DRO 93] DROGOUL A., 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, 1993.
- [FAG 95] FAGIN R., HALPERN J.Y., MOSES Y., VARDI M.Y., *Reasoning about Knowledge*, MIT Press, 1995.

- [FER 89] FERBER J., « Eco Problem Solving: how to solve a problem by interactions », dans *Proceedings of 9th workshop on Distributed Artificial Intelligence*, 1989.
- [FER 91] FERBER J., JACOPIN E., « The framework of Eco Problem Solving », dans Demazeau Y., Müller J.P. (dir.), *Decentralized Artificial Intelligence II, Proceedings of the 2nd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-90)*, Elsevier North-Holland, p. 103-114, 1991.
- [FER 92] FERGUSON I.A., *TouringMachines: An Architecture for dynamic, Rational, Mobile Agents*, PhD Thesis, Clare Hall, University of Cambridge, Grande-Bretagne, 1992.
- [FER 95a] FERBER J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, 1995.
- [FER 95b] FERGUSON I.A., « Integrated control and coordinated behaviour: a case for agent models », dans Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents: Theories, Architectures and Languages*, p. 203-218, Springer-Verlag, LNAI n° 890, 1995.
- [FER 97] FERRAND N., DEMAZEAU Y., BAEIJS C., « Systèmes multi-agents réactifs et résolution de problèmes spatialisés », *Revue d'Intelligence Artificielle*, vol. 12, n° 1, p. 37-72, 1997.
- [FIR 89] FIRBY R.J., *Adaptive execution in dynamic domains*, PhD Thesis, Yale University, 1989.
- [FIS 95a] FISCHER K., MÜLLER J.P., PISCHEL M., SCHIER D., « A Model for cooperative transportation scheduling », dans *Proceedings 1st International Conference on Multi-Agent Systems*, p. 109-116, 1995.
- [FIS 95b] FISCHER K., MÜLLER J.P., PISCHEL M., « A Pragmatic BDI Architecture », dans Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents: Theories, Architectures, and Languages*, p. 203-218, Springer-Verlag, LNAI n° 890, 1995.
- [FIS 96] FISCHER K., MÜLLER J.P., PISCHEL M., « AgenDA – A General Testbed for Distributed Artificial Intelligence Applications », dans *Foundations of Distributed Artificial Intelligence*, O'Hare G.M.P., Jennings N.R. (dir.), John Wiley & Sons, p. 401-427, 1996.
- [FIS 97] FISHER M., WOOLDRIDGE M.J., « On the formal specification and verification of multi-agent systems », *Int. Journal of Cooperative Information Systems*, vol. 6, n° 1, p. 37-65, 1997.
- [FIS 99] FISHER M., « Representing Abstract Agent Architecture », dans Müller J.P., Singh M.P., Rao A.S. (dir.), *Intelligent Agents: Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1555, p. 227-242, 1999.
- [FOD 83] FODOR J.A., *The modularity of Mind*, MIT Press, Cambridge, MA, 1983.
- [FRA 96] FRANKLIN S., GRAESSER A., « Is it an Agent, or just a Program? A Taxonomy for autonomous agents », Müller J.P., Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents III: Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1193, p. 21-35, 1996.

- [GEN 94] GENESERETH M.R., KETCHPEL S.P., « Software Agents », *Communications of the ACM*, p. 48-53, juillet 1994.
- [GEO 87] GEORGEFF M.P., LANSKY A.L., « Reactive reasoning and planning », dans *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, p. 677-682, 1987.
- [GEO 89] GEORGEFF M.P., INGRAND F.F., « Decision-making in an embedded reasoning system », dans *Proceedings of the International Joint Conference on Artificial Intelligence*, p. 972-978, 1989.
- [GEO 99] GEORGEFF M.P., PELL B., POLLACK M., TAMBE M., WOOLDRIDGE M.J., « The Belief-Desire-Intention Model of Agency », dans Müller J.P., Singh M.P., Rao A.S. (dir.), *Intelligent Agents: Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1555, 1999.
- [GMY 95] GMYTRASIEWICZ P.J., DURFEE E.H., « A rigorous operationnal formalization of recursive modeling », dans *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS)*, AAAI Press, p. 125-132, 1995.
- [GOO 93] GOODWIN R., Formalizing properties of agents, Tech. Report CMU-CS-93-158, Carnegie Mellon University, 1993.
- [GUE 96] GUESSOUM Z., Un environnement opérationnel de conception et de réalisation de systèmes multi-agents, Thèse de doctorat, Université Pierre et Marie Curie, Paris, 1996.
- [GUI 99] GUILLEMET A., HAÏK G., MEURISSE T., BRIOT J.P., LHUILLIER M., « Mise en œuvre d'une approche componentielle pour la conception d'agents », dans Gleizes M.P., Marcenac P. (dir.), *Actes des journées francophones IAD et SMA*, Hermès, p. 53-66, 1999.
- [HAD 95] HADDADI A., « Towards a pragmatic theory of interactions », dans *Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS)*, AAAI Press, p. 133-139, 1995.
- [HAN 99] HANNOUN M., BOISSIER O., SICHMAN J.S., SAYETTAT C., « MOISE : un modèle organisationnel pour la conception de systèmes multi-agents », dans Gleizes M.P., Marcenac P. (dir.), *Actes des journées francophones IAD et SMA*, Hermès, p. 105-118, 1999.
- [HAS 92] HASSOUN M., DEMAZEAU Y., LAUGIER C., « Motion control of a car-like robot: potential field and multi-agent approaches », dans *Proceedings of 5th IEEE International conference on Intelligent Robots and Systems*, 1992.
- [HAU 94] HAUGENEDER H., STEINER D., MCCABE F.G., « IMAGINE: A framework for building multi-agent systems », dans Deen S.M. (dir.), *Proceedings of the 1994 International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, p. 31-64, 1994.
- [HAY 85] HAYES-ROTH B., « A Blackboard Architecture for Control », *Artificial Intelligence*, n° 26, p. 251-321, 1985.
- [HAY 95] HAYES-ROTH B., PFLEGER K., LALANDA P., MORIGNOT P., BALABANOVIC M., « A Domain-specific software architecture for adaptative intelligent systems », *IEEE transaction on Software Engineering*, vol. 21, n° 4, p. 288-301, 1995.

- [HUH 97] HUHNS M., SINGH M.P., « Agents and multi-agent systems: themes, approaches and challenges », dans Huhns M., Singh M.P. (dir.), *Readings in agents*, Morgan Kaufmann, p. 1-24, 1997.
- [INV 97] D'INVERNO M., KINNY D., LUCK M., WOOLDRIDGE M.J., « A Formal Specification of dMARS », dans Singh M.P., Rao A., Wooldridge M.J. (dir.), *Intelligent Agents IV: Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1365, p. 155-176, 1997.
- [JEN 93a] JENNINGS N.R., « Commitments and Conventions: the foundation of coordination in multi-agent systems », dans *Know. Eng. Rev.*, vol. 8, n° 3, p. 223-250, 1993.
- [JEN 93b] JENNINGS N.R., « Specification and implementation of a Belief-Desire-Joint-Intention Architecture for collaborative problem solving », *Int. Journal of Intelligent and Cooperative Information Systems*, vol. 2, n° 3, p. 289-318, 1993.
- [JEN 95] JENNINGS N.R., « Controlling cooperative problem solving in industrial multi-agent systems using joint intentions », *Artificial Intelligence*, vol. 75, n° 2, p. 195-240, 1995.
- [JEN 96] JENNINGS N.R., MAMDANI E.H., LARESGOITI I., PEREZ J., CORERA J., « Using ARCHON to develop real-world DAI applications », *IEEE Expert*, vol. 11, n° 6, p. 64-70, 1996.
- [JEN 00a] JENNINGS N.R., NORMAN T.J., FARATIN P., O'BRIEN P., ODGERS B., « Autonomous Agents for Business Process Management », dans *International Journal of Applied Artificial Intelligence*, vol. 14, n° 2, p. 145-189, 2000.
- [JEN 00b] JENNINGS N.R., FARATIN P., NORMAN T.J., O'BRIEN P., ODGERS B., ALTY J.L., « Implementing a Business Process Management System using ADEPT: A Real-World Case Study », dans *International Journal of Applied Artificial Intelligence*, vol. 14, n° 5, p. 421-465, 2000.
- [JUN 99] JUNG C.G., « On the Role of Computational Models for specifying Hybrid Agents », dans *Cybernetics and Systems*, p. 749-754, 1999.
- [KAE 86] KAEHLING L.P., « An architecture for intelligent reactive systems », dans Georgeff M.P., Lansky A.L. (dir.), *Reasoning about Actions and Plans – proceedings of the 1986 Workshop*, Morgan-Kaufmann, p. 395-410, 1986.
- [KEN 97] KENDALL E.A., MALKOUN M.T., JIANG C., « Multi-Agent System design based on object oriented patterns », *Report on Object Oriented Design in Journal of Object Oriented Programming*, p. 41-47, février 1997.
- [LES 98] LESSER V.R., « Reflections on the Nature of Multi-Agent Coordination and its Implications for an Agent Architecture », *Jour. Autonomous Agents and Multi-Agent Systems*, 1, p. 89-111, 1998.
- [LUC 95] LUCK M., D'INVERNO M., « A formal framework for Agency and Autonomy », dans *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS)*, AAAI Press, p. 254-260, 1995.
- [MAE 90a] MAES P., *Designing Autonomous Agents*, MIT Press, 1990.

- [MAE 90b] MAES P., « Situated agents can have goals », dans MAES P. (dir.), *Designing Autonomous Agents*, MIT Press, p. 49-70, 1990.
- [MAR 97] MARCENAC P., « Modélisation d systèmes complexes par agent », *Technique et science informatiques*, vol. 16, n° 8, p. 1013-1037, 1997.
- [MIL 99] MILOJICIC D., « Mobile Agent Applications », *IEEE Concurrency*, p. 80-90, juillet-septembre, 1999.
- [MON 97] MONROE R.T., KOMPANEK A., MELTON R., GARLAN D., « Architectural styles, Design patterns and objects », *IEEE Software*, p. 43-52, 1997.
- [MOO 90] MOORE R.C., « A formal theory of knowledge and action », dans Allen J.F., Hendler J., Tate A. (dir.), *Readings in Planning*, p. 480-519, Morgan Kaufmann, 1990.
- [MOU 96] MOULIN B., CHAIB-DRAA B., « An Overview of Distributed Artificial Intelligence », dans O'Hare G.M.P., Jennings N.R. (dir.), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, p. 3-55, 1996.
- [MUL 94] MÜLLER J.P., PISCHEL M., « An architecture for dynamically interacting agents », *Int. J. Intell. Coop. Inf. Syst.*, vol. 3, n° 1, p. 25-45, 1994.
- [MUL 95] MÜLLER J.P., PISCHEL M., THIEL M., « Modelling reactive behaviour in vertically layered agent architectures », dans Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents: Theories, Architectures, and Languages*, p. 261-276, Springer-Verlag, LNAI n° 890, 1995.
- [MUL 96] MÜLLER J.P., *The Design of Intelligent Agents: A Layered Approach*, Springer-Verlag, LNAI n° 1177, 1996.
- [MUL 97] MÜLLER J.P., « Control Architectures for Autonomous and Interacting Agents: A Survey », Cavedon L., Rao A.S., Wobcke W. (dir.), *Intelligent Agent Systems: theoretical and practical issues*, p. 1-26, Springer-Verlag, LNAI n° 1209, 1997.
- [MUL 99] MÜLLER J.P., *The right agent (architecture) to do the right thing*, *Intelligent Agents V*, Springer-Verlag, LNAI 1555, p. 213-225, 1999.
- [NAK 98] NAKASHIMA H., NODA I., « Dynamic subsumption architecture for programming intelligent agents », dans *Proceedings of 3rd International Conference on Multi-Agent Systems*, p. 190-197, 1998.
- [NEW 90] NEWELL A., « *Unified Theories of Cognition* », Harvard University Press, Cambridge/Londres, 1990.
- [OCC 97] OCCELLO M., DEMAZEAU Y., « Vers une approche de conception et de description réursive en univers multi-agents », dans *Proceedings of Journées Francophones sur Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, p. 143-157, 1997.
- [PAT 87] PATTISON H.E., CORKILL D.D., LESSER V.R., « Instantiating Description of Organizational Structures », Huhns M.N. (dir.), *DAI, Research Notes in AI*, Morgan Kaufmann/Pitman Publishers, p. 59-96, 1987.
- [RAO 91] RAO A.S., GEORGEFF M.P., « Modeling rational agents within a BDI-Architecture », dans Fikes R., Sandewall E. (dir.), *Proceedings of Knowledge Representation and Reasoning*, p. 473-484, 1991.

- [RAO 92] RAO A.S., GEORGEFF M.P., « An Abstract Architecture for Rational Agents », dans Rich C., Swartout W., Nebel B. (dir.), *Proceedings of Knowledge Representation and Reasoning*, Morgan/Kaufmann, p. 439-449, 1992.
- [RAO 93] RAO A.S., GEORGEFF M.P., « A model-theoretic approach to the verification of situated reasoning systems », dans *Proceedings of 13th Int. Joint Conf. on Artificial Intelligence*, p. 318-324, 1993.
- [RAO 95] RAO A.S., GEORGEFF M.P., « BDI Agents: from theory to practice », dans *Proceedings of 1st International Conference on Multi-Agent Systems (ICMAS)*, AAAI Press, p. 312-319, 1995.
- [RAO 96] RAO A.S., « AgentSpeak(L): BDI Agents speak out in a logical computable language », dans Van De Velde W., Perram J. (dir.), *Agents Breaking Away: Proceedings of 7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, p. 42-55, Springer-Verlag, LNAI n° 1038, 1996.
- [RAO 99] RAO A.S., GEORGEFF M.P., SINGH M.P., « Formal methods in DAI », dans G. Weiss (dir.), *Multi-Agent systems: A modern approach to Distributed Artificial Intelligence*, MIT Press, p. 331-376, 1999.
- [ROS 86] ROSENSCHEIN S.J., KAEHLING L.P., « A Situated view of representation and contro », dans Halpern J.Y. (dir.), *Proceedings of the 1986 conference on theoretical aspects of reasoning about knowledge*, p. 83-98, Morgan Kaufmann, 1986.
- [RUS 91] RUSSEL S.J., WEFALD E., *Do the right thing – Studies in Limited Rationality*, MIT Press, Cambridge, MA, 1991.
- [RUS 95] RUSSEL S.J., NORVIG P., *Artificial Intelligence: A Modern Approach*, Prentice-Hall, 1995.
- [SHA 95] SHAW M., GARLAN D., *Formulations and Formalisms in Software Architecture*, Springer-Verlag, LNCS n° 1000, 1995.
- [SHO 93] SHOHAM Y., « Agent Oriented Programming », *Artificial Intelligence*, vol. 60, n° 1, Elsevier, p. 51-92, 1993.
- [SHO 95] SHOHAM Y., TENNENHOLZ M., « Social laws for artificial agent societies : off-line design », *Artificial Intelligence*, 73, p. 231-252, Elsevier, 1995.
- [SIC 95] SICHMAN JS., DEMAIZEAU Y., « Exploiting social reasoning to deal with agency level inconsistency », dans *Proceedings of the 1st International Conference on Multi-Agent Systems (ICMAS'95)*, AAAI Press, p. 352-359, 1995.
- [SIE 97] SIERRA C., JENNINGS N.R., NORIEGA P., PARSONS S., « A framework for argumentation-based negotiation », dans Singh M.P., Rao A.S., Wooldridge M.J. (dir.), *Intelligent Agents IV: Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1365, p. 177-192, 1997.
- [SIN 94] SINGH M.P., *Multi-Agent systems: a theoretical framework for intentions, know-how, and communications*, Springer-Verlag, LNAI n° 799, 1994.
- [STE 90] STEELS L., « Cooperation between Distributed Agents through Self-Organization », dans Demazeau Y., Müller J.-P. (dir.), *Decentralized AI – Proceedings of the 1st*

European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89), Elsevier North-Holland, p. 175-196, 1990.

- [STE 96] STEINER D.D., « IMAGINE: An Integrated Environment for Constructing Distributed Artificial Intelligence Systems », dans O'Hare G.M.P., Jennings N.R. (dir.), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, p. 345-364, 1996.
- [THO 95] THOMAS S.R., « The PLACA Agent Programming Language », dans Wooldridge M.J., Jennings N.R. (dir.), *Intelligent Agents I: Agent Theories, Architectures and Languages*, Springer-Verlag, p. 355-369, 1995.
- [VER 00] VERCOUTER L., Conception et mise en œuvre de systèmes multi-agents ouverts et distribués, Thèse de doctorat, Ecole nationale supérieure des mines de Saint-Etienne, Université Jean Monnet, décembre 2000.
- [VER 90] VERE S., BICKMORE T., « A basic agent », *Computational Intelligence*, n° 6, p. 41-60, 1990.
- [VER 98] VERHAREN E.M., DIGNUM F., BOS S., « Implementation of Cooperative Agent Architecture Based on the Language-Action Perspective », dans Singh M.P., Rao A.S., Wooldridge M.J. (dir.), *Intelligent Agents IV: Agent Theories, Architectures, and Languages*, Springer-Verlag, LNAI n° 1365, p. 31-44, 1998.
- [WAV 92] WAVISH P., *Exploiting Emergent Behaviour in Multi-Agent Systems*, dans Werner E., Demazeau Y. (dir.), *Decentralized AI III – Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, Elsevier North-Holland, p. 297-310, 1992.
- [WIT 92] WITTIG T., *ARCHON: An Architecture for Multi-Agent Systems*, Ellis Horwood, Chichester, 1992.
- [WOO 92] WOOLDRIDGE M.J., The logical modelling of computational multi-agent systems, PhD Thesis, UMIST, Dept. of computation, Manchester, Grande-Bretagne, 1992.
- [WOO 99] WOOLDRIDGE M.J., « Intelligent Agents », dans Weiss G. (dir.), *Multi-Agent systems: A modern approach to Distributed Artificial Intelligence*, MIT Press, 1999.

Chapitre 3

Modèles de communication

3.1. Introduction

Dans le domaine de la recherche, la *communication* est un objet d'étude multiforme exploré par maintes disciplines, comme la psychologie, la biologie, les sciences cognitives ou encore l'informatique et, en ce qui nous concerne, les systèmes multi-agents (SMA). Chaque discipline aborde la communication selon un axe privilégié, sa problématique propre et renvoie à des objets d'étude divers. Il n'y a donc pas une mais des sciences de la communication et, par nature, le domaine de la communication est pluridisciplinaire.

Rien d'étonnant donc que le domaine des SMA, pour lequel les notions d'interaction et de communication entre agents sont fondamentales, voire premières par rapport à la notion même d'agent, s'inspire, emprunte, importe des théories issues des autres disciplines. Nous verrons en particulier qu'à l'instar des recherches en sciences de la communication qui ont compris depuis longtemps que la communication ne se réduit pas à la simple transmission d'une information porteuse d'un sens que le récepteur décode, les recherches en SMA se sont orientées vers des modèles plus complexes de communication entre agents. Par exemple, distinguer la signification du message de l'intention de communication de l'émetteur est devenu une proposition classique en sciences humaines qui influence aujourd'hui la façon de voir la communication entre agents artificiels. Ainsi, les modèles de communication entre agents artificiels tendent à s'éloigner du modèle « idéal » développé en 1948 par l'ingénieur Claude E. Shannon [SHA 48], selon lequel la communication se réduit à la transmission d'une information d'un émetteur à un

récepteur par le biais d'un canal, modèle qui constitue le modèle dominant en informatique (envoi de message entre objets, appel de procédures à distance, etc.).

La communication est d'autant plus un élément central des systèmes multi-agents que les agents ne peuvent rarement se contenter d'effectuer leurs actions sans considérer ce que les autres agents sont en train de faire. Il est alors nécessaire de faire appel à la coordination des actions des agents. Cette coopération est rendue possible par la communication. L'agent est forcé de communiquer avec les autres et d'obtenir l'engagement de leur part que certaines actions seront menées ou certains buts réalisés.

Ainsi, même si la discipline des SMA est récente, elle a maintenant organisé de manière autonome le développement d'un savoir et d'un savoir-faire dans le domaine de la communication entre agents et, plus particulièrement, en matière de langages d'interaction et de protocoles d'interaction. C'est ce savoir et ce savoir-faire que nous développons dans la suite de ce chapitre.

On retrouvera dans ce chapitre, de façon relativement interchangeable, les deux termes *communication* et *interaction* que l'on rencontre dans la littérature. On peut néanmoins préciser dans cette introduction succinctement leur sens. Communiquer signifie « être en relation avec » (selon le dictionnaire Le Robert). Plus particulièrement, dans les systèmes multi-agents, ce terme fait référence à une transmission d'information entre agents. Interagir, selon le dictionnaire, veut dire « avoir une action réciproque ». En termes de systèmes multi-agents, l'interaction comprend donc la communication ainsi que l'action sur le monde. Les deux termes se rejoignent (ce qui fait que nous les utiliserons indifféremment) car la transmission d'information induit parfois chez l'autre un comportement spécifique, et donc dans ce cas, communiquer signifie 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 de l'agent 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 [FER 97].

Pour traiter de la communication entre agents logiciels, le présent chapitre dresse tout d'abord un panorama des modes de communication possibles : de la communication minimale à la communication par passage de messages. Ensuite, un paragraphe est dédié aux langages de communication. Nous y développons dans un premier temps les notions générales de la théorie des actes de langage. Dans un deuxième temps les deux principaux langages, ACL et KQML, sont présentés. Enfin, tout un paragraphe est consacré à la structuration des échanges d'information par protocoles. Cette approche est décrite de manière générale et les différentes représentations des protocoles sont traitées. Certains protocoles d'interaction majeurs, souvent rencontrés dans la littérature spécialisée, sont ensuite passés en revue. Ce chapitre se conclut sur deux perspectives : d'une part, les travaux récents qui prennent en compte les derniers développements sur les actes de langage et,

d'autre part, une nouvelle approche de la structuration de l'interaction qui est appelée *règles de conversation*.

Les notions connexes à celles de communication et d'interaction entre agents, comme les notions de conflit, de négociation, de coordination ou encore de coopération sont quant à elles abordées dans le chapitre 4.

3.2. Modes de communication entre agents

Nous distinguons essentiellement deux modes de communication : la communication indirecte qui est une communication par signaux *via* l'environnement, et la communication directe, qui procède à un échange de messages entre les agents. Ce second cas est celui considéré dans ce chapitre. Nous donnons sa définition et précisons ses implications sur l'agent et le système multi-agent dans cette section. Mais préalablement, nous présentons rapidement les grandes lignes du premier mode.

3.2.1. Communication indirecte *via* l'environnement

Dans ce cas, la communication est restreinte à un ensemble fini de signaux ayant des interprétations fixées. Cette classe de communication s'adresse à des agents de type réactif¹. Ces agents n'ont pas de communication « explicite ». Celle-ci a lieu par propagation de stimuli *via* l'environnement tels les phéromones pour les fourmis, et uniquement par des actions sur l'environnement, par exemple la consommation de ressources de l'environnement [DRO 98], mais sans communication ajoutée.

Ce mode de communication autorise une coordination limitée. En effet, une coopération sophistiquée est difficilement réalisable car cela nécessite d'avoir un très grand nombre de types de signaux qui se réfèrent directement à une action. Des actions complexes ne peuvent pas non plus être réalisées puisqu'il n'existe pas de syntaxe des signaux pour construire ces actions. Par conséquent, des commandes arbitrairement complexes, des requêtes et des intentions ne peuvent pas être exprimées.

Dans le reste de ce chapitre nous ne nous intéresserons plus à la communication entre agents réactifs. Ce type de communication est également appelé la

1. Il ne faut pas confondre un agent de type réactif avec un agent de type cognitif qui a la propriété d'être réactif. Cette propriété là signifie que l'agent doit être capable de réagir au changement de son environnement.

communication réactive. Le lecteur est invité à se reporter au chapitre 1 pour une présentation des agents réactifs.

3.2.2. *Communication directe par passage de messages*

Le mode de communication entre agents par passage de messages provient initialement du domaine des acteurs [HEW 77]. Il permet à un agent de planifier un acte de communication envers un autre agent. On parle de communication intentionnelle.

Cette approche s'inspire de l'interaction sociale telle que nous la trouvons dans d'autres contextes comme la communication entre humains (voir paragraphe 3.3.1). Par conséquent, les mécanismes et les termes utilisés supportent cette vision de l'interaction et de la communication de haut niveau. A la différence des autres approches, celle-ci a donné lieu à des théories calculatoires formelles permettant de savoir comment communiquer incrémentalement ou comment ajuster un plan de communication pour s'adapter à un monde incertain où les circonstances changent, rendant ainsi possible la communication complexe d'informations stratégiques.

Dans la littérature, nous pouvons distinguer deux façons de répondre à cette approche. Premièrement, cela consiste pour l'agent à bâtir son propre plan de communication au moment où il en a besoin pour réaliser sa tâche. Dans [STE 93] par exemple, les agents ont des primitives de coopération représentées par des plans qui font appel à des fonctions minimales de communication basées sur certains actes de langage.

La seconde, plus répandue maintenant, s'appuie sur la notion de protocole pour les séquencements des messages qu'un agent peut envoyer. Nous l'étudierons dans la section 3.4. Les agents concernés par ce mode de communication sont de type cognitif. Ce type d'agent est celui considéré dans le reste de ce chapitre. Nous définissons ci-après quelles sont les caractéristiques qu'ils doivent présenter du point de vue de la communication.

3.2.3. *Communication par passage de messages et agents*

La dichotomie faite parfois entre agents cognitifs et agents réactifs fonctionnant par un mécanisme de perception/action sur le monde est artificielle. Les agents cognitifs peuvent eux aussi être amenés à fonctionner selon ce mode. Nous pouvons dire qu'il existe un *continuum* entre ces deux modes extrêmes de communication ou les agents correspondants sont plus ou moins cognitifs [DEM 91].

Certaines exigences sur le comportement d'un agent cognitif sont communément adoptées, et leur langage de communication à base d'actes de langages vise à

supporter et faciliter ces comportements. L'agent peut communiquer avec d'autres agents, il a des capacités de communication et par là même d'envoi, de réception et d'interprétation des messages.

Du point de vue de l'interaction, on suppose généralement que les agents constituent un système coopératif dans le sens où les agents cherchent à travailler ensemble et tentent toujours de déclarer clairement leur attitude en réponse au discours des autres [NUM 90]. En matière de passage de messages, nous pouvons trouver deux versions. La version synchrone ne permet pas de *bufferiser* les messages, donc un message à remettre à un agent a besoin d'un rendez-vous entre l'agent émetteur et l'agent destinataire. La version asynchrone suppose que la *bufferisation* des messages est possible, donc il n'est pas nécessaire que l'agent destinataire donne son accord avant qu'un message lui soit envoyé. C'est notamment l'approche adoptée par les acteurs [AGH 86].

Le modèle de communication abordé dans ce chapitre ne s'intéresse pas aux services des niveaux bas et intermédiaires souvent associés à la communication entre des systèmes logiciels distribués, tels que les protocoles de réseaux, les services de la couche transport du modèle OSI, etc. En effet, nous supposons l'existence de ces services utilisés pour convoyer les séquences d'octets que comprend la communication interagent.

Par exemple, pour la communication entre agents par protocoles d'interaction (voir section 3.4) nous supposons que la couche transport sous-jacente offre une distribution des messages en séquence, précise, asynchrone, point à point, sujette à des délais finis, ou une notification d'échec de la livraison [KIN 98].

Le modèle de la FIPA [FIP 99] (voir paragraphe 3.3.2.2) supporte les hypothèses ci-dessus. En raison de l'asynchronisme, la séquence observée de transmission et de réception peut différer pour deux agents. Les protocoles doivent être conçus pour assurer que toutes les collisions qui peuvent légalement survenir (c'est-à-dire à l'exclusion de celles causées par la violation de protocoles) soient bénignes et que, là où cela est possible et utile, les parties en interaction puissent détecter de telles situations.

Nous ne traiterons pas dans ce chapitre des aspects temporels de la communication, car ils sont peu présents dans la littérature. Dans cette rubrique, nous pouvons néanmoins citer l'outil METATEM [FIS 94]. Cet aspect est parfois abordé sous l'angle de la logique temporelle ou des langages formels comme dans [SIN 92] qui se base sur CTL* (une logique des prédicats avec des opérateurs temporels) augmentée des prédicats de l'intention et du savoir-faire et d'opérateurs.

3.3. De l'échange des connaissances aux langages

C'est dans les années 1960 que les philosophes du langage, spécialistes de la *pragmatique*, ont mis en évidence que l'unité minimale de la communication humaine est l'acte de langage (*speech act*). Ainsi, chaque énonciation constitue une véritable action socialement réglée et dépendant de conditions de succès. Dès lors, se développa un ensemble de recherches ayant pour point de départ cette idée selon laquelle l'utilisation de formes langagières est une action à part entière [AUS 62]. Searle en 1969 [SEA 69] proposa une première catégorisation des actes de langage, puis Searle et Vanderveken proposèrent en 1985 [SEA 85] une axiomatisation de la théorie initiale.

Les langages d'interaction entre agents sont inspirés de ces travaux que nous présentons brièvement dans le paragraphe suivant avant de présenter les spécifications du langage KQML (*Knowledge Query and Manipulation Language*), qui est sans doute le plus connu et sans conteste à l'origine des travaux actuels de l'organisation FIPA². Cette organisation a pour objectif de proposer des spécifications pour les agents et, pour ce qui nous préoccupe, des spécifications pour un langage d'interaction entre agents dans le but louable de maximiser l'interopérabilité des applications à base d'agents. Ce langage spécifié s'appelle ACL (*Agent Communication Language*) et nous le décrivons au paragraphe 3.3.2.2.

3.3.1. La philosophie du langage ordinaire

3.3.1.1. La notion d'acte de langage

Austin participa au développement de ce qu'il est convenu d'appeler la *philosophie du langage ordinaire*. Dans son ouvrage fondateur *How to do things with words* [AUS 62, [AUS 70], il a établi que toute profération d'un énoncé en contexte, toute énonciation, permet la réalisation d'un acte qui, en tant que tel, vise à accomplir, faire, quelque chose. Cet acte est appelé *acte de langage*.

Il s'est intéressé à rechercher l'élément spécifique d'un tel acte de langage. Cet élément est la valeur (force) d'illocution qui représente le « faire » essentiel d'une énonciation (une requête, une déclaration, une promesse, etc.). Il a mis en évidence deux autres éléments, la valeur de locution (la *signification*, au sens classique du terme) et la valeur de perlocution (les conséquences – les effets – sur les autres, sur soi). Une distinction nette est alors posée entre l'acte de dire quelque chose (l'acte locutoire), l'acte effectué en disant quelque chose (l'acte illocutoire) et l'effet obtenu

2. La FIPA (*Foundation for Intelligent Physical Agents*) est une organisation internationale à but non lucratif, créée en 1996, qui a pris en charge le problème de la standardisation des technologies agents. Il s'agit d'une initiative industrielle. [Http://www.fipa.org](http://www.fipa.org).

en disant quelque chose (acte perlocutoire). Par exemple, en disant : « Tu ne dois pas faire cela » (locutoire), je conseille ou j'ordonne (illocutoire) et dissuade ou importune autrui (perlocutoire).

Pour Austin, la valeur d'illocution est fondamentalement la plus importante. Pour simplifier, disons que cette valeur est pour lui la *clé* de l'utilisation et de la compréhension du langage naturel. Il précise que cette valeur est spécifiquement conventionnelle (au sens que l'acte illocutoire doit être produit conformément à une convention), tandis que les deux autres éléments, la valeur de locution et la valeur de perlocution, ne le sont pas.

Comme nous le verrons dans la suite, les langages d'interaction entre agents utilisent aujourd'hui cet aspect primordial de toute énonciation. Il est même d'usage de réduire l'acte de langage à sa seule composante illocutoire et il est courant de dire acte illocutoire en lieu et place d'acte de langage.

3.3.1.2. La théorie des actes de langage

La théorie des actes de langage a été véritablement « fondée » à la suite des travaux d'Austin, par Searle dans son ouvrage intitulé *Speech Acts* [SEA 69]. Puis suivirent de nombreux travaux, en particulier ceux de Searle et Vanderveken [SEA 85, VAN 88, VAN 90, VAN 91, VAN 99] ou encore de Trognon et Brassac [BRA 92, TRO 91, TRO 92]. Nous ne retiendrons de ces recherches, que les résultats qui nous permettent de mieux appréhender les travaux sur les langages d'interaction entre agents.

Un résultat important a consisté à établir que tout acte de langage « porte » un acte illocutoire qui appartient à l'une des cinq catégories suivantes :

- acte *assertif* : le locuteur exprime comment les objets auxquels il se réfère sont dans le monde. Il s'agit des assertions, des informations, des témoignages, des démentis, etc. ;
- acte *commissif* : le locuteur s'engage à accomplir une action. Il s'agit des promesses, des vœux, des menaces, etc. ;
- acte *directif* : le locuteur fait en sorte que l'interlocuteur accomplisse une action. Il s'agit des demandes, des questions, des ordres, des conseils, etc. ;
- acte *expressif* : le locuteur manifeste son état mental face à un état de chose. Il s'agit des excuses, remerciements, félicitations, récriminations, etc. ;
- acte *déclaratif* : le locuteur accomplit au moment de l'énonciation l'action qu'il dit accomplir. Il s'agit des définitions, des condamnations, des ratifications, etc.

Ces actes sont formalisables sous la forme F(P), avec F la force illocutoire s'appliquant à P, le contenu propositionnel. Ainsi, les énonciations « Est-ce qu'il pleut ? » et « Est-ce que tu vas bien ? » ont *a priori* la même force illocutoire

(question) mais des contenus propositionnels différents ; les énonciations « Pourvu qu'il fasse beau demain ! » et « Il fera beau demain » ont des contenus propositionnels identiques mais des forces illocutoires *a priori* différentes puisque la première semble exprimer un vœu alors que la seconde est plutôt une assertion.

Le contenu propositionnel P est une proposition qui décrit l'état du monde (le fait qu'il fasse beau pour l'affirmation « Il fait beau ») ou l'action que le locuteur s'engage à réaliser (le fait d'être à midi à la gare pour le locuteur pour la promesse « Je serai à la gare à midi »). P est la composante *représentationnelle* de l'acte de langage alors que la force illocutoire F est la composante *intentionnelle*. La force illocutoire F a été décomposée en six composantes primaires qui rassemblent toutes les caractéristiques d'une énonciation. La plus importante de ces composantes est le but (ou intention) poursuivie par le locuteur.

A partir des valeurs possibles des composantes de la force illocutoire F, une taxonomie des verbes dits *performatifs* (*to perform* en anglais signifie accomplir ou faire) a été proposée. Ces verbes sont un sous-ensemble des verbes d'une langue, ce sont ceux qui permettent de décrire l'action réalisée par le locuteur. Ces verbes ont été classés selon qu'ils permettent d'accomplir un acte assertif, directif, déclaratif, commissif ou expressif. Parmi les verbes performatifs français, citons : penser, affirmer, dire, informer, etc. (verbes assertifs) ; demander, avertir, réclamer, exiger, supplier, etc. (verbes directifs) ; déclarer, stipuler, ratifier, etc. (verbes déclaratifs) ; promettre, acquiescer, garantir, refuser, etc. (verbes commissifs) ; féliciter, excuser, approuver, déplorer, etc. (verbes expressifs). En revanche, les verbes écrire, laver, marcher, conduire, jouer, imaginer, etc., ne sont pas des verbes performatifs.

Les verbes performatifs permettent de paraphraser « performativement » les énoncés. Ainsi, tout énoncé peut être exprimé par :

Je « verbe performatif » « contenu propositionnel »

Par exemple, l'énoncé « As-tu un vélo ? » peut être paraphrasé par « Je *demande* si tu as un vélo » ou encore l'énoncé « Oui, j'ai un vélo » peut être paraphrasé par « J'*affirme* que j'ai un vélo ».

Les recherches réalisées autour de la théorie des actes de langage vont bien au-delà de la simple classification des actes de langage et des verbes performatifs. En particulier, l'étude des relations logiques qui relient les propriétés d'un acte ou de plusieurs est tout à fait centrale dans ce courant de recherche. Toutefois, c'est l'aspect *taxonomie* qui a été majoritairement exploité pour la conception des langages de communication entre agents et nous arrêtons donc là, sciemment, notre exposé.

3.3.2. *Langages de communication*

3.3.2.1. *Le langage KQML (Knowledge Query and Manipulation Language)*

Le langage KQML est un langage de communication conçu pour l'échange, entre logiciels (agents), des informations, des connaissances ou encore des services. Il a été proposé en 1993 par le consortium DARPA-KSE (*Knowledge Sharing Effort*)³ dont l'objectif est de développer et promouvoir des standards pour le partage et la réutilisation de bases de connaissances et de systèmes à base de connaissances. KQML a donc été conçu dans ce souci de développer des techniques et outils génériques pour favoriser l'interopérabilité d'agents hétérogènes.

Les primitives du langage KQML sont appelées des *performatifs* et, comme le terme le suggère pleinement, ce langage trouve ses fondements dans la théorie des actes de langage que nous avons présentée brièvement. Les performatifs, au sens de KQML, définissent les actions (les opérations) que les agents peuvent faire pour communiquer les uns avec les autres. Ainsi, un message KQML est constitué essentiellement d'un verbe performatif associé à un contenu dont le format de représentation n'est pas imposé par le langage ; ce peut être un message au format KIF (*Knowledge Interchange Format*) [GEN 98], un message en PROLOG ou encore une chaîne de caractères. Nous noterons bien le souci des concepteurs de dissocier le contenu propositionnel du message, de la valeur d'illocution du message.

3.3.2.1.1. La syntaxe du message KQML

Conceptuellement, un message KQML se décompose en trois niveaux :

- un niveau *message*, qui permet d'identifier le type d'acte (nom du performatif), le langage et l'ontologie utilisés et qui s'appliquent au contenu ;
- un niveau *communication*, qui permet d'identifier l'émetteur, le récepteur et le message ;
- un niveau *contenu*.

La syntaxe générale du message KQML est présentée par la figure 3.1. Selon le performatif utilisé, tous ces paramètres ne sont pas nécessaires.

3. <http://www.cs.umbc.edu/kse/>

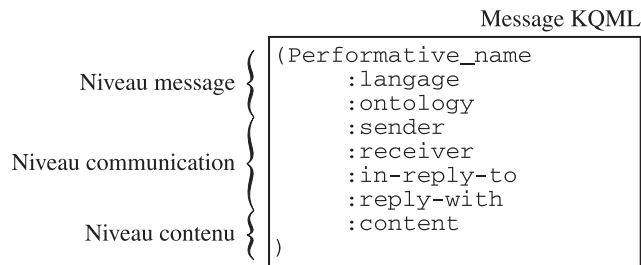


Figure 3.1. Syntaxe générale d'un message KQML

Prenons l'exemple souvent employé d'un agent A désirant connaître quels sont les parents de John. Pour cela, il interroge un agent B possédant cette information. La requête est la suivante :

```

(stream-all
:language      Prolog
:ontology      Genealogy
:sender        A
:receiver      B
:in-reply-to   id0
:reply-with    id1
:content       « parent(John,X) »)
  
```

L'agent A veut que l'agent B l'informe de façon exhaustive (*all*) à propos des parents de John. L'usage de *stream-all* (plutôt que *ask-all*) spécifie que les éléments de réponses doivent être délivrés un par un. Voici, ci-après, une des réponses de l'agent B, lequel informe l'agent A que la mère de John s'appelle Alice. Cette réponse sera suivie par une deuxième du même type (pour indiquer le père de John) et un message de fin de type *eos* (*end-of-stream*).

```

(tell
:language      Prolog
:ontology      Genealogy
:sender        B
:receiver      A
:in-reply-to   id1
:reply-with    id2
:content       « parent(John,Alice) »)
  
```

3.3.2.1.2. Les performatifs KQML

KQML a évolué entre 1993 [FIN 93], année de la première version des spécifications, et 1997, année de la publication d'une proposition pour de nouvelles spécifications [LAB 97]. Ainsi, les performatifs ont évolué au cours du temps. Nous retiendrons ici, pour présenter les performatifs, la classification proposée dans les dernières spécifications, même s'il ne s'agit pas des spécifications officielles mais d'une proposition. En effet, cette classification nous semble être rigoureuse et avoir le mérite de la clarté puisque d'un ensemble initial de 41 performatifs répartis en 11 types, nous trouvons dans la proposition de 1997 une taxonomie de 36 performatifs en seulement 3 catégories. Sans prétendre à l'exhaustivité, nous donnons ici quelques éléments à propos de ces 3 catégories de performatifs.

Les performatifs de discours (*discourse performatives*) sont au nombre de 18 et peuvent être considérés comme étant les plus proches de la théorie des actes de langage. Ils sont utilisés pour échanger des informations et des connaissances entre les agents. A travers ces performatifs, un agent peut par exemple demander des informations à un autre agent. Il a la possibilité de donner, si elles existent, les informations présentes dans sa base de connaissances ou de démentir la présence d'une information dans celle-ci. Ou encore, un agent peut demander qu'un autre agent insère une information dans sa base de connaissances. D'autres performatifs ont un rôle plus particulier, par exemple pour indiquer qu'un agent est capable de traiter certains messages KQML. Les performatifs *ask-if*, *ask-one*, *stream-all*, *deny*, *tell*, *insert*, etc., sont quelques exemples.

Les performatifs d'interconnexion (*networking and facilitation performatives*) sont au nombre de 11 (*register*, *unregister broadcast*, *broker*, etc.). Ils aident à la mise en relation des agents entre eux. Ils sont réservés (à deux exceptions près) aux communications entre agents (du système multi-agent) et des agents particuliers appelés *facilitators*. En effet, une des particularités de KQML est d'utiliser, en plus des agents proprement dit, ces agents *facilitators* qui « facilitent » la communication et fournissent des fonctions, telles que l'association entre les adresses physiques et noms symboliques, la tenue d'un registre des agents et de leurs compétences.

Les performatifs d'exception (*intervention and mechanics of conversation performatives*) sont au nombre de 7 (*error*, *sorry*, *standby*, etc.). Ils sont utiles pour changer le déroulement normal des échanges ou pour terminer prématurément les échanges.

3.3.2.2. Le langage ACL (Agent Communication Language)

Le langage ACL de la FIPA [FIP 99] est fondé également sur la théorie des actes de langage et a bénéficié grandement des résultats de recherche de KQML. Si l'approche du langage ACL est globalement semblable à celle de KQML, un grand soin a été apporté aussi bien à la description formelle de la sémantique des actes de

communication qu'à l'introduction de protocoles régissant les règles d'échanges de messages.

Pour présenter le langage, nous avons choisi de mettre essentiellement l'accent sur la description formelle de la sémantique des actes de communication. Les protocoles seront, quant à eux, abordés essentiellement au paragraphe 3.4.3.

3.3.2.2.1. La syntaxe du message ACL

La syntaxe du message est assez similaire à celle de KQML. La figure 3.2 résume sur un exemple les éléments principaux du message ACL.

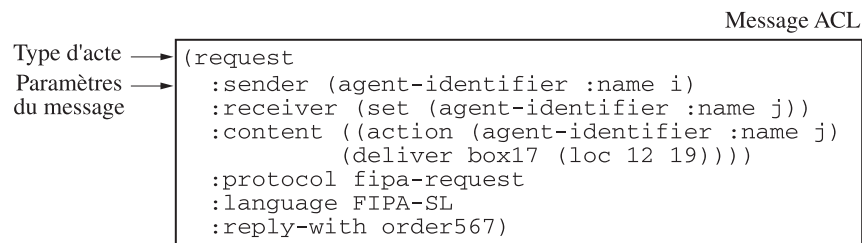


Figure 3.2. Exemple d'un message ACL

3.3.2.2.2. La sémantique des actes de communication ACL

Les actes de communication sont définis par une description en langage naturel, et également formellement en langage SL (*Semantic Language*). Dans le cadre de ce chapitre nous ne donnons que quelques éléments sur le langage SL pour comprendre dans quel « esprit » sont définis formellement les actes de communication.

SL est un langage qui s'appuie sur une logique modale du premier ordre (pour des précisions sur cette logique, voir [SAD 91a]). Ce cadre logique est similaire sur bien des aspects à celui proposé par Cohen et Levesque [COH 90b] et définit un modèle mental de l'agent en trois attitudes : *belief*, *uncertainty* et *choice* (ou, dans une certaine mesure, *goal*), formalisées respectivement par trois opérateurs modaux *B*, *U* et *C*. Ainsi :

- $B_i p$: « l'agent *i* croit (implicitement) que la proposition *p* est vraie » ;
- $U_i p$: « l'agent *i* est incertain à propos de *p* mais pense que *p* est plus vraisemblable que $\neg p$ » ;
- $C_i p$: « l'agent *i* souhaite que la proposition *p* soit vérifiée ».

Afin de raisonner également en termes d'actions (notées a_i), des séquences d'actions ($a_1 ; a_2$ par exemple) et des choix entre actions (a_1/a_2 par exemple) peuvent être exprimés ; les opérateurs *Feasible*, *Done* et *Agent* sont également définis. Ainsi :

- *Feasible* (a, p) : « L'action a peut être effectuée et si tel est le cas la proposition p sera alors vraie » ;
- *Done* (a, p) : « L'action a a été effectuée et la proposition p est vraie depuis » ;
- *Agent* (i, a) : « L'agent i est le seul effectuant ou allant effectuer l'action a ».

Sont également définis à partir des concepts précédents, les concepts de *persistent goal* ($PG_i p$ signifie que l'agent i a p comme but persistant – but pour lequel l'agent est engagé) et d'*intention* ($I_i p$ signifie que l'agent i a l'intention de réaliser p).

3.3.2.2.3. Les catégories d'actes ACL

Le langage ACL différencie les *actes primitifs* des *actes composés* construits à partir de ces premiers. Le format général de tout acte, est le suivant :

$\langle i, \text{act}(j, p) \rangle$

- avec :
- i , l'agent émetteur,
 - act , l'acte de communication,
 - j , l'agent receveur,
 - p , la proposition.

A chaque acte sont en outre associés :

- les préconditions qui représentent les conditions qui doivent être vraies pour que l'acte puisse être utilisé ; ces préconditions sont notées FP (*Feasibility Preconditions*) ;
- les effets attendus après utilisation de l'acte ; ces effets sont notés RE (*Rational Effects*).

Sont définis quatre actes primitifs : *inform*, *request*, *confirm*, *disconfirm*. Ils peuvent être accomplis directement par un agent, planifiés par un agent ou encore demandés par un agent à un autre agent. A titre d'exemple, les actes *inform* et *request* sont définis ainsi :

inform : $\langle i, \text{inform}(j, \phi) \rangle$

FP: $B_i \phi \wedge \neg B_i (B_i \neg \phi \vee \bigvee_j B_j \neg \phi)$

RE: $B_j \phi$

l'agent i informe l'agent j que la proposition ϕ est vraie

(avec : $B_i \neg \phi \equiv B_i \phi \vee B_i \neg \phi$)

request : $\langle i, \text{request}(j, a) \rangle$

FP: $FP(a) [i|j] \wedge Bi \text{ Agent}(j, a) \wedge \neg Bi \text{ Ij Done}(a)$

RE: $Done(a)$

l'agent i demande à l'agent j de réaliser l'action a .

(avec :

a est l'action

$FP(a)$ définit les préconditions sur a ;

$FP(a) [i|j]$ définit la partie des préconditions sur a qui représente les attitudes mentales de l'agent i .)

A partir des quatre actes primitifs est construit un ensemble d'actes composés. Par exemple, l'acte *query_if* permet de formuler des actes comme : « Je vous demande (*request*) de me dire (*inform-if*) s'il pleut ».

query_if: $\langle i, \text{query-if}(j, \phi) \rangle \equiv$

$\langle i, \text{request}(j, \langle j, \text{inform-if}(i, \phi) \rangle) \rangle$

FP: $\neg Bi \phi \wedge \neg Ui \phi \wedge \neg Bi \text{ Ij Done}(\langle j, \text{inform-if}(i, \phi) \rangle)$

RE: $Done(\langle j, \text{inform}(i, \phi) \rangle | \langle j, \text{inform}(i, \neg \phi) \rangle)$

L'agent i demande à j qu'il l'informe si ϕ est vrai ou non.

(avec :

inform_if: $\langle i, \text{inform-if}(j, \phi) \rangle \equiv$

$\langle i, \text{inform}(j, \phi) \rangle | \langle i, \text{inform}(j, \neg \phi) \rangle$

FP : $Bi \phi \wedge \neg Bi (Bi \neg \phi \vee Ui \neg \phi)$

RE : $Bi \phi$)

Les actes composés sont au nombre de 18 ; on trouve par exemple : *request_when*, *request_whenever*, *refuse*, *reject_proposal*, *propose*, etc.

Enfin, cinq catégories répartissent l'ensemble des vingt-deux actes primitifs et composés, en fonction de la finalité de chaque acte :

- transmettre une information,
- demander une information,
- négocier,
- accomplir une action,
- gérer un problème.

3.3.2.2.4. Les protocoles d'interaction

Les spécifications d'ACL [FIP 99] incluent outre la spécification du langage, une description d'un ensemble de protocoles d'interaction entre agents comme le très connu *Contract Net Protocol* (CNP) [SMI 80], qui est un des premiers protocoles de

communication de haut niveau pour la résolution de problèmes distribués. D'autres protocoles comme les protocoles de vente aux enchères ou encore de négociation complètent les spécifications. Ces protocoles sont décrits précisément dans les spécifications de la FIPA ; ils seront abordés au paragraphe 3.4.3.

3.3.2.3. *En résumé*

KQML et ACL sont les deux langages phares actuels. Ils se rejoignent sur l'idée première et fondamentale de la philosophie du langage ordinaire, à savoir que toute énonciation est un acte (une action) ; cet acte vise à accomplir (faire) quelque chose. C'est pourquoi aussi bien les messages ACL que KQML expriment une valeur d'illocution (les primitives des langages appelées performatifs pour KQML et actes de communication en ACL) qui s'applique à un contenu (le champ *:content* dans les deux cas). Leur différence essentielle provient des aspects théoriques liés à la sémantique des langages qui s'ancre dans des théories de l'agent quelque peu différentes.

Par le biais du langage ACL, nous avons vu quelques aspects de la sémantique formelle des langages d'interaction. Durant les dix dernières années, de nombreux travaux ont été effectués dans cette direction. Les plus connus sont certainement ceux de Cohen et Perrault [COH 79] et Cohen et Levesque [COH 90a]. Ces derniers s'appuient sur leur théorie de l'intention [COH 90b] pour développer une théorie selon laquelle les actes de langage sont des actions à part entière qu'accomplissent des agents rationnels pour satisfaire leurs intentions.

Un des reproches fait au langage KQML [COH 95] a d'ailleurs concerné son manque de formalisme pour exprimer la sémantique des actes, jusqu'à ce qu'une sémantique formelle soit proposée [LAB 96, LAB 98]. Elle repose sur la définition, pour chaque acte, de préconditions, de postconditions et de conditions d'arrêt. En quelques mots et sans entrer dans les détails, ces conditions décrivent les états des agents en termes d'attitudes mentales (*knowledge, belief, desire, intention*). Le langage utilisé pour décrire les états des agents ne conditionne pas le modèle théorique de l'agent mais contraint toutefois la façon dont les attitudes mentales peuvent être combinées pour définir les différents états de l'agent.

D'autres langages ont contribué de près ou de loin à faire évoluer ce domaine de connaissance ; citons pour mémoire le langage ARCOL [SAD 91b] de France Télécom, qui est à la base des spécifications d'ACL, le langage COOL [BAR 95], ou encore le langage IL [DEM 95], qui a été un des premiers langages à intégrer explicitement la notion de protocole d'interaction.

3.4. Structuration des échanges : les protocoles d'interaction

Dans cette section, nous décrivons tout d'abord le principe de la communication entre agents selon un protocole et nous présentons ensuite les types de spécifications

majeurs utilisés pour représenter et exploiter ces protocoles d'interaction. Enfin, nous présentons les principaux protocoles.

3.4.1. *Description de l'approche par protocole*

Dans les SMA, atteindre la coordination entre agents autonomes est un problème majeur. En général, les agents coordonnent leurs actions en échangeant des messages. Un processus de coordination est une séquence de ces interactions entre agents.

Une sémantique pour des actes de langage individuels (ou performatifs) n'est pas suffisante pour pleinement supporter l'interaction entre agents : le contenu de l'acte de langage est important, tout comme la capacité à formuler des dialogues (ou des protocoles, comme le bien connu *Contract Net*) qui guident l'interaction entre agents.

Traditionnellement, un protocole est une spécification de ces interactions. Il précise qui peut dire quoi à qui et les réactions possibles à ce qui est dit par l'autre partie. Une spécification syntaxique formelle [WER 89] est donnée en définissant les types de messages qu'il est possible d'envoyer. On définit informellement la sémantique des messages ainsi que les stratégies des rôles, en décrivant les procédures qui déterminent les réactions d'un agent à un message donné. Ces réactions peuvent inclure la génération de nouveaux messages. Les protocoles sont particulièrement utiles pour des agents qui n'incorporent pas explicitement la sémantique formelle des actes de langage.

La fonction de base d'un protocole d'interaction est de fournir aux agents une façon de communiquer utilement sans avoir à planifier explicitement chaque acte de langage en se basant sur ses effets attendus. Quand nous utilisons un protocole d'interaction, nous supposons que le concepteur du protocole a effectué une analyse hors ligne suffisante et garantit ainsi que suivre le protocole permettra d'atteindre les buts associés aux états finaux identifiés du protocole. Plus un protocole est efficace, moins d'informations ont besoin d'être transmises et moins de temps est passé dans la communication.

Les agents participent à différents protocoles en interagissant de manière appropriée entre eux, par exemple en répondant aux messages, en effectuant des actions dans leur domaine respectif, ou en mettant à jour leurs états locaux. Les protocoles peuvent alors être pris comme un moyen de spécifier la politique que les agents vont suivre dans leurs interactions avec les autres ; par exemple, cette politique déterminerait les conditions sous lesquelles une requête pourra être satisfaite ou une permission accordée ou une affirmation crue.

3.4.2. Représentation et spécification des protocoles

Afin d'arriver à une approche unifiée pour définir et développer adéquatement des protocoles d'interaction, nous avons besoin d'une représentation expressive pour esquisser les dialogues. Plus essentiellement, la représentation doit aussi être immédiatement traduisible en une implémentation équivalente. Ensuite, l'émetteur et le destinataire d'un message, qui initie une forme de coopération, devraient être capables d'utiliser le même protocole. Cela semble utile du fait que la relation des actions effectuées par les participants au dialogue au cours de l'interaction est explicitement représentée et parce que cela permet une représentation plus compacte du dialogue. A cet égard, la représentation doit aussi prendre en compte le fait que les deux participants d'un dialogue utiliseront une copie du même protocole.

Les nœuds du graphe représentent les états du dialogue, les arcs représentent les transitions d'un état conversationnel à l'autre, et les branches (suites de transitions) représentent des transitions possibles (voir figure 3.3).

Les nœuds adjacents représentent alternativement l'agent qui a émis le premier message du protocole puis l'agent destinataire de ce message. De façon duale, pour un agent qui suit un protocole les nœuds représentent alternativement des états d'émission ou de réception de message.

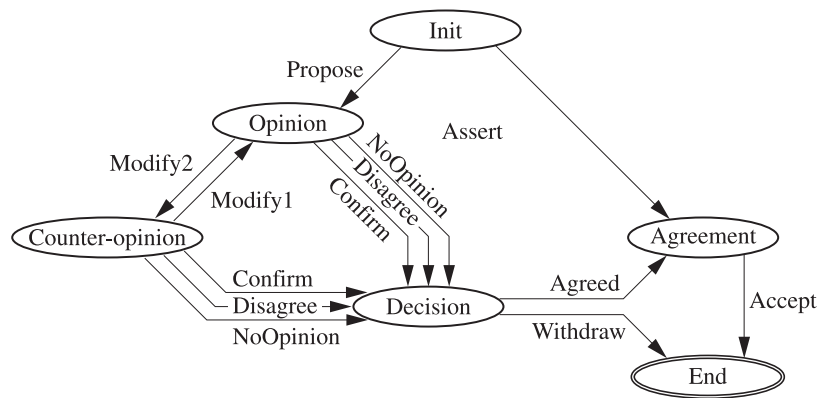


Figure 3.3. Graphe d'état du protocole de SIAN

Graphe état-transition

Un protocole peut être représenté par un graphe dirigé dans lequel les nœuds dénotent les états du protocole et les arcs, qui sont étiquetés par des modèles de

messages qui consistent en des types (actes de langage) et des restrictions sur le contenu, dénotent des transitions entre ces états. Un protocole peut par conséquent être défini comme un ensemble de paires composées d'une transition et d'un modèle de message, où chaque transition est représentée par une paire d'états (initial, final). Il existe un ou plusieurs chemins entre le nœud initial et tous les autres nœuds. Sur tous ces chemins, les arcs sont respectivement étiquetés des types de messages en réception puis en émission. La plupart des protocoles sont classiquement spécifiés à l'aide de machines d'états finis. Nous en mentionnons quelques-uns au paragraphe 3.4.3.

T. Winograd et F. Flores [WIN 86] représentent et analysent les dialogues par des réseaux d'actes de langage. En ce qui concerne la représentation, leurs graphes sont très expressifs et faciles à suivre. Cependant, il leur manque l'information nécessaire pour les rendre opérationnels.

La représentation adoptée dans [BUR 95] peut être vue comme une abstraction des diagrammes état-transition, où l'envoi, la réception et le traitement d'un message, sont attachés aux nœuds du graphe. En plus de représenter des états conversationnels, ces nœuds peuvent aussi être des appels à des sous-protocoles. [BUR 95] présente une méthode modulaire de construction de protocoles d'interaction à partir éventuellement de protocoles prédéfinis. Cette approche ressemble à celle d'AgentTalk évoquée ci-dessous, bien qu'elle soit moins complète, ou à celle du protocole de négociation SANP [CHA 92] qui emploie une représentation modulaire similaire en permettant aux nœuds d'un protocole d'être soit des états conversationnels simples soit des appels à d'autres protocoles (appelés *phases*). Une machine à états finis est utilisée pour décrire les protocoles de coordination en AgentTalk [KUW 95]. Il s'agit d'une machine à états finis étendue qui autorise des variables et qui est souvent utilisée dans le domaine des protocoles [HOL 91]. Les états du protocole de coordination sont ainsi explicitement définis et les actions d'un agent peuvent être aisément définies pour chaque état.

Graphe de raisonnement

J. Campbell et M. d'Iverno [CAM 90] proposent, une représentation graphique générale dans laquelle les protocoles non seulement montrent la communication entre deux agents, mais aussi les états du raisonnement. Bien que ces graphes puissent remplir leur but, l'adoption d'une telle notation comme moyen général pour représenter les protocoles pose potentiellement des problèmes car nous pouvons nous demander si le raisonnement associé à chaque nœud est purement indépendant de l'application. En effet, les processus de raisonnement et de décision impliqués au cours d'une interaction sont dépendants de l'application, et par conséquent peuvent difficilement être présents au niveau de la représentation.

Réseaux de Petri

T. Kreifelts et F. Von Martial [KRE 91] commencèrent avec des diagrammes similaires et ensuite les remplacèrent par des réseaux de Petri pour pouvoir résoudre le problème du dépassement et du croisement des messages. Malheureusement, la puissance d'expression des diagrammes originaux est perdue et les composantes et les étapes du protocoles ne sont plus visibles. Le nouveau diagramme est davantage un diagramme général décrivant la communication entre deux agents. On retombe sur le problème de la communication physique et on ne se situe plus au niveau protocole d'interaction.

Plus récemment, nous trouvons [KON 98] et [COS 99], qui modélisent la conversation entre agents à l'aide de réseaux de Petri.

Langage formel

Un défaut de l'approche par diagramme états-transitions est qu'elle ne capture pas adéquatement les contraintes qui s'étalent sur différents niveaux de protocoles. Par conséquent, certains chercheurs [DIV 98] sont en train de développer des spécifications formelles complètes de protocoles dans le langage de spécification Z.

La spécification d'un protocole d'interaction peut être présentée en utilisant le langage Z qui est de plus en plus employé dans le domaine de la recherche sur les agents [DIV 97]. Z est un langage de spécification formel orienté modèle basé sur la théorie des ensembles et la logique du premier ordre [SPI 92]. Les composants-clés d'une spécification Z sont les définitions de l'espace d'état d'un système et des opérations possibles qui font passer d'un état à un autre.

Langages de description de protocoles

Plusieurs langages de description de protocoles d'interaction ont également été définis.

Un protocole représenté par une machine d'états finie peut être codé textuellement selon une structuration PDL (*Protocol Description Language*) introduite dans [POP 93] et étendu dans [BOR 99]. Ce langage est suffisamment universel pour décrire un certain nombre de protocoles qui ont été présentés dans la littérature. Pour cela, il répertorie pour un protocole donné les classes d'agents concernés, les états et les transitions.

Le langage COOL décrit les protocoles par des règles de transitions d'états en s'appuyant sur des messages KQML [BAR 95].

Le langage Cool (un autre) décrit les protocoles par des plans multi-agents qui consistent en des primitives de communication et des procédures que les agents doivent suivre [KOL 95].

Le langage AgenTalk décrit des protocoles en incorporant un mécanisme d'héritage [KUW 95] comme pour les langages orientés objets. Il permet de concevoir incrémentalement de nouveaux protocoles à partir de protocoles existants, et les concepteurs d'applications peuvent implémenter des protocoles adaptés pour convenir à chaque application en s'appuyant sur des protocoles existants indépendants du domaine.

Pour décrire la coopération dans les systèmes multi-agents, [DEG 92] a inventé un langage de spécification général qui est transformable en un langage déjà implémenté. Le but de ce langage est de s'abstraire des détails qui n'ont pas de rapport avec les agents qui interagissent, de telle manière que les concepteurs peuvent aisément définir la connaissance d'interaction.

3.4.3. Principaux protocoles d'interaction

Pour terminer cette section consacrée aux protocoles d'interaction nous faisons ici un rapide tour d'horizon des principaux protocoles définis dans la littérature. Nous passons également en revue quelques domaines où ceux-ci ont été utilisés. Faute de place, un inventaire exhaustif des protocoles ne peut pas être dressé mais nous avons essayé de les regrouper en familles, car en effet, certains protocoles majeurs ont donné lieu à de nombreuses variantes.

3.4.3.1. Contract Net Protocol

La plus grande et la plus connue des familles de protocoles relève du réseau contractuel *Contract Net Protocol* (CNP) [SMI 80]. Il s'agit d'un des premiers protocoles de communication de haut niveau pour la résolution de problèmes distribués. Il a été repris à de nombreuses occasions et adapté tant pour des applications précises que pour la définition de systèmes multi-agents. Initialement, ce protocole a été appliqué à l'allocation de ressources ainsi qu'à des systèmes de surveillance distribuée tel le contrôle du trafic aérien [SMI 81]. Il a été généralisé pour prendre en compte des contraintes globales [CON 91]. Il a également été étendu en introduisant la gestion de l'incertitude qui réside dans tout processus de négociation et a été appliqué au commerce électronique [SAN 95]. Dans [MAL 88], il a donné lieu à un protocole d'ordonnancement distribué pour le système Entreprise qui est une application du protocole par réseau de contrat pour l'allocation de tâches entre stations de travail connectés sur un réseau local.

En fait, la quasi-totalité des publications scientifiques qui se rapportent un tant soit peu au thème de l'interaction dans les SMA positionnent leur approche par rapport à celle du réseau contractuel. Nous allons succinctement présenter ci-dessous son principe. Pour une étude plus fouillée de ce protocole, le lecteur est invité à consulter les articles de référence [SMI 80, SMI 81] ou le chapitre 7 de [FER 95].

L'intérêt de ce protocole est qu'il permet à un groupe d'agents : (1) d'interagir pour résoudre un problème global, (2) de se partager dynamiquement la charge de travail, et (3) d'intégrer leurs résultats. Il fournit également un ensemble de types de messages indiquant les sortes d'informations que les agents doivent s'échanger afin d'effectuer une certaine forme de coopération.

Un agent peut demander de l'aide lorsqu'il est confronté, soit à une tâche trop importante pour lui tout seul, soit à une tâche pour laquelle il n'a aucune capacité. Si la tâche est trop importante, il tentera d'abord de la partitionner en sous-tâches gérables puis tentera de trouver d'autres agents qui ont les capacités appropriées pour gérer les nouvelles tâches. Si la tâche initiale va au-delà de ses capacités, il tente d'emblée de trouver un autre agent plus approprié pour la traiter.

Dans les deux cas, le problème de l'agent revient à en trouver d'autres dont les capacités correspondent à la tâche qu'il souhaite distribuer. S'il connaît les autres agents ayant l'expertise nécessaire, il peut leur signaler directement. S'il n'a pas connaissance d'autres agents qui puissent l'assister (ou si la tâche ne requiert aucune expertise particulière), il peut simplement décrire la tâche à l'ensemble des agents.

Si un autre agent disponible pense être capable de réaliser la tâche, il informe l'agent initial de sa disponibilité et indique éventuellement les capacités particulières qu'il peut avoir. L'agent initial peut alors se retrouver avec plusieurs agents volontaires et choisir entre eux. L'agent finalement choisi peut alors demander des informations supplémentaires et les deux peuvent alors engager une communication plus directe pour la durée de la tâche.

Afin de distribuer la charge de travail à un ensemble d'agents, ceux qui ont des tâches à exécuter doivent trouver d'autres agents capables d'exécuter ces tâches. En même temps, les agents non occupés ont à trouver des tâches qui leur conviennent sur lesquelles travailler.

3.4.3.2. *Protocoles d'introduction*

Afin de développer des applications multi-agents dans des systèmes évolutifs, il est nécessaire de répondre à des questions du type : comment des agents peuvent-ils s'informer l'un l'autre de leur existence, leurs compétences et leurs désirs ? Tout agent doit être capable d'informer les autres agents avec lesquels il voudrait travailler coopérativement dans quelle situation et comment il peut l'accéder, parce qu'il n'est pas raisonnable de supposer que tous les autres agents le connaissent.

L'enregistrement (ou introduction) est un processus par lequel des demandeurs potentiels de services sont capables d'acquérir des informations sur les services offerts par les fournisseurs, incluant leur identité et les signatures de leurs types de paramètres. [KIN 98] propose un tel protocole d'enregistrement.

E. Durfee et T. Montgomery [DUR 90] décrivent comment une hiérarchie de comportements peut être utilisée dans un protocole pour permettre aux agents de découvrir et résoudre souplesment des interactions. Les agents qui ne savent pas initialement avec qui interagir utilisent cette hiérarchie pour échanger des abstractions sur leur comportement envisagé.

S. Berthet *et al.* [BER 92] proposent des protocoles permettant à des agents d'informer les autres sur leurs capacités dans le cadre de systèmes multi-agents ouverts. Ceux-ci ont été appliqués en robotique pour le développement d'un système de vision général.

3.4.3.3. *Autres protocoles*

Nous évoquons ici divers autres protocoles.

Avec le *protocole d'apprentissage coopératif* basé sur l'adaptation [SIA 91], chaque agent utilise ses informations locales pour inférer des hypothèses partielles et ensuite interagir avec les autres agents pour à la fois corriger et augmenter la confiance dans ses hypothèses. Ce protocole a été étendu pour prendre en compte l'imprécision dans l'évaluation des préférences sur les hypothèses, et a été appliqué à la supervision de réseaux de télécommunications [KON 95].

Le *protocole de négociation multiniveau* SANP [CHA 92] est basé sur la notion d'agent « attaquant » et d'agent « défenseur ». Lorsqu'une issue ne peut pas être trouvée, les agents demandent à une autorité supérieure de trancher leur différend. Il existe également un agent ayant un rôle d'arbitre. Une conversation peut être reportée à une date ultérieure s'il n'y a pas suffisamment d'information pour continuer.

Le *protocole d'avertissement* de [CAM 90] permet à des agents de s'informer mutuellement de sources de danger perçues par un autre.

Par *protocole de vente aux enchères* on fait classiquement référence à deux types de protocoles qui s'inspirent de deux techniques opposées. Pour la vente aux enchères anglaise, l'acheteur propose initialement un prix inférieur au prix du marché et augmente graduellement le montant. Pour la vente aux enchères hollandaise, le commissaire priseur commence par annoncer un prix élevé au-dessus de la valeur marchande du produit et descend jusqu'à ce qu'un acheteur se déclare intéressé. Les protocoles de cette famille sont très utilisés dans le domaine du commerce électronique [ROD 99].

Il existe également tout un tas de *protocoles élémentaires* qui permettent à un agent de demander simplement à un autre d'effectuer une action (*request protocol*), d'effectuer une action dans le futur (*request when protocol*), de demander une

information (*query protocol*), etc. Ces protocoles sont très souvent utilisés, et de manière implicite. Ils sont recensés dans [FIP 99] et spécifiés en ACL.

3.5. Conclusions et perspectives

Nous avons vu dans ce chapitre que les modèles d'interaction dans les systèmes multi-agents s'inspirent des théories de la communication humaine. Des travaux sur la modélisation des interactions entre agents, plus récents que ceux présentés ici, ont été entrepris en s'intéressant aux derniers résultats des chercheurs dans la lignée de la philosophie du langage ordinaire. Le livre *Analyse et simulations de conversations – de la théorie des actes de discours aux systèmes multi-agents* est intéressant à cet égard [MOU 99]. Citons également les travaux de Chaib-draa et Vanderveken [CHA 99] qui s'appuient sur les notions de succès et de satisfaction d'un acte [VAN 91]. D'autres travaux [BRA 99, CHI 98] se fondent sur des modèles de la conversation humaine qui intègrent des aspects non littéraux et adopte une perspective dynamique relative à la mise en séquence des actes de langage (approche *dialogique*) s'intéressant ainsi essentiellement à l'enchaînement conversationnel [BRA 92]. D'autre part, la dimension sociale dans l'interaction est de plus en plus prise en compte [BOU 99, CAS 96, VON 99]. S'enrichissant de ces aspects dynamiques et de « règles sociales », les interactions et communications entre agents évoluent donc actuellement pour tendre à devenir des « conversations », se rapprochant ainsi des caractéristiques inhérentes de la communication humaine.

Au niveau de la spécification des protocoles d'interaction, les formalismes présentés dans ce chapitre ne s'attachent pas tant à spécifier des règles d'interaction que des modèles computationnels en faisant appel typiquement à des automates ou des formalismes de réseaux de transition. Les règles de décision individuelles pour lesquelles les protocoles d'interaction traditionnels ont été conçus sont laissées implicites dans ces formalismes. Cela soulève des questions importantes dans la pratique actuelle de la conception des protocoles d'interaction. Par exemple, comment garantir que les protocoles sont interopérables ou peuvent être composés, comment les modifier, et comment modéliser et vérifier les effets attendus quand on suit ces règles ?

Récemment, la littérature sur les protocoles d'interaction a évolué vers la notion de règles de conversation (*conversation policies* en anglais) dont le but est d'imposer un nombre de contraintes sur ce que les agents peuvent dire, interpréter ou faire durant une conversation [GRE 99]. Il s'agit d'un ensemble de spécifications déclaratives qui gouvernent la communication entre des agents logiciels qui utilisent un langage de communication interagent. On est ainsi en mesure de prendre en considération plusieurs hypothèses importantes concernant les interactions entre agents qui étaient typiquement laissées implicites jusque-là dans les descriptions de protocoles d'interaction, et de les séparer des contraintes de séquençement qui sont

traditionnellement le sujet des protocoles d'interaction. On distingue deux types de contraintes : d'une part, les contraintes syntaxiques qui restreignent les séquences possibles de messages et les états conversationnels, et, d'autre part, les contraintes sémantiques qui définissent quelles sont les actions externes au protocole qu'un agent peut effectuer lors de certains états conversationnels ou quand il reçoit certains messages. Parmi les travaux qui correspondent à cette optique, on peut citer [PIT 99] qui permet non seulement de spécifier des règles de conversation mais également de définir les langages de communication associés. Le système POS [OUD 00], quant à lui, s'appuie sur la sémantique opérationnelle structurée des langages de programmation [HEN 90] qui permet de prendre en compte la spécification de *patterns d'états* conversationnels et de *patterns de messages*. On aboutit ainsi à des systèmes ayant une puissance d'expression universelle autorisant le traitement de n'importe quel protocole complexe. Une telle approche permet ainsi de séparer proprement la spécification d'un protocole de l'architecture interne des agents.

3.6. Bibliographie

- [AGH 86] AGHA G., *Actors: A Model of Concurrent Computation, in Distributed Systems*, MIT Press, Cambridge, MA, 1986.
- [AUS 62] AUSTIN J.L., *How to do things with words*, Clarendon Press, Oxford, 1962.
- [AUS 70] AUSTIN J.L., *Quand dire c'est faire*, Editions du Seuil, Paris, 1970.
- [BAR 95] BARBUCEANU M., FOX M.S., « COOL: A Language for Describing Coordination in MultiAgent System », dans *First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, USA, juin 1995, AAAI Press, p. 17-24, 1995.
- [BER 92] BERTHET S., DEMAZEAU Y., BOISSIER O., « Knowing Each Other Better », dans *11th International Workshop on Distributed Artificial Intelligence (IWDAI-92)*, Glen Arbor, Michigan, février 1992.
- [BOR 99] BORDINI R.H., *Contributions to an Anthropological Approach to the Cultural Adaptation of Migrant Agents*, PhD thesis, University College London, Department of Computer Science, février 1999.
- [BOU 99] BOUZOUA K., MOULIN B., « Connaissances implicites et sociales : dialogisme des interactions discursives », dans *Analyse et Simulation de conversations: De la théorie des actes de discours aux systèmes multiagents*, Textes rassemblés par B. Moulin, S. Delisle et B. Chaib-draa, L'Interdisciplinaire de Lyon, 1999.
- [BRA 92] BRASSAC C., « Analyse de conversations et théorie des actes de langage », *Cahiers de Linguistique Française*, 13, p. 62-76, 1992.
- [BRA 99] BRASSAC C., PESTY S., « Simuler la conversation : un défi pour les systèmes multi-agents », dans *Analyse et Simulation de conversations : De la théorie des actes de discours aux systèmes multiagents*, Textes rassemblés par B. Moulin, S. Delisle et B. Chaib-draa, L'Interdisciplinaire de Lyon, 1999.

- [BUR 95] BURMEISTER B., HADDADI A., SUNDERMEYER K., « Generic, Configurable, Cooperation Protocols for Multi-Agent Systems », dans Castelfranchi C. et Muller J.-P. (dir.), *From Reaction to Cognition*, Springer-Verlag, LNAI 957, Berlin, Allemagne, p. 157-171, 1995.
- [CAM 90] CAMPBELL J., D'INVERNO M., « Knowledge Interchange Protocol », dans Demazeau Y., Müller J.-P. (dir.), *Decentralized AI*, vol. I, Elsevier Science Publishers B.V., Amsterdam, p. 63-80, 1990.
- [CAS 96] CASTELFRANCHI C., CONTE R., « Distributed Artificial Intelligence and social science : critical issues », dans O'Hare G., Jennings N. (dir.), *Foundations of Distributed Artificial Intelligence*, Wiley, 1996.
- [CHA 92] CHANG M.K., WOO C.C., « SANP: A Communication Level Protocol for Negotiations », dans Werner E. et Demazeau Y. (dir.), *Decentralized AI*, vol. III, Elsevier Science Publishers B.V., Amsterdam, p. 31-54, 1992.
- [CHA 99] CHAIB-DRAA B., VANDERVEKEN D., « Agent Communication Language: Towards a Semantics based on Success, Satisfaction, and recursion », dans Müller J.P., Singh M.P., Rao A.S. (dir.), *Intelligent Agents V, Agent Theories, Architectures and Language*, Springer-Verlag, LNAI, Berlin, 1999.
- [CHI 98] CHICOISNE G., RICORDEL PM., PESTY S., DEMAZEAU Y., « Outils et pistes pour la pratique du dialogisme entre agents », dans Barthès, Chevrier, Brassac (dir.), *Systèmes Multi-Agents : de l'Interaction à la socialité, Actes des 6ème Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Pont-à-Mousson, p. 163-176, 1998.
- [COH 79] COHEN P.R., PERRAULT C.R., « Elements of Plan-Based Theory of Speech Acts », *Cognitive Science*, vol. 3, n° 1, p. 177-212, 1979.
- [COH 90a] COHEN P.R., LEVESQUE H.J., « Rational Interaction as the basis for communication », dans Cohen P.R., Morgan J., Pollack M.E. (dir.), *Intentions in Communication*, MIT Press, p. 221-256, 1990.
- [COH 90b] COHEN P.R., LEVESQUE H.J., « Intention is choice with commitment », *Artificial Intelligence*, vol. 42, n° 2-3, p. 213-361, 1990.
- [COH 95] COHEN P.R., LEVESQUE H.J., « Communicative actions for artificial agents », *Proceedings of the First International Conference on Multi-agent Systems (ICMAS-95)*, AAAI Press/MIT Press, p. 65-72, 1995.
- [CON 91] CONRY S. E., KUWABARA K., LESSER V. R., MEYER R. A., « Multistage Negotiation for Distributed Constraint Satisfaction », *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 21, n° 6, p. 1462-1477, 1991.
- [COS 99] COST R.S., CHEN Y., FININ T., LABROU Y., PENG Y., « Modeling Agent Conversation with Colored Petri Nets », dans Bradshaw J. (dir.), *Autonomous Agents'99, Special Workshop on Conversation Policies*, mai 1999.
- [DEG 92] DEGREEF P., CLARK K., MCCABE F., « Toward a Specification Language for Cooperation Methods », dans *GWAI-92 Workshop Supporting collaborative Work Between Human Experts and Intelligent Cooperative Information Systems*, 1992.

- [DEM 91] DEMAZEAU Y., MULLER J.-P., « From Reactive to Intentional Agents », dans *Decentralized A.I., Vol. II*, Elsevier Science Publishers B.V., Amsterdam, p. 3-10, 1991.
- [DEM 95] DEMAZEAU Y., « From Interactions to Collective Behaviour in Agent-Based Systems », dans *First European Conference on Cognitive Science*, Saint-Malo, France, p. 117-132, avril 1995.
- [DIV 97] D'INVERNO M., KINNY D., LUCK M., WOOLDRIDGE M., « A Formal Specification of dMARS », dans *Intelligent Agents IV: Fourth International Workshop on Agent Theories, Architectures, and Languages (ATAL-97)*, Springer-Verlag, LNAI 1365, 1997.
- [DIV 98] D'INVERNO M., KINNY D., LUCK M., « Interaction Protocols in Agentis », dans Demazeau Y. (dir.), *Third International Conference on MultiAgent Systems (ICMAS-98)*, Paris, France, IEEE, p. 112-119, juillet 1998.
- [DRO 98] DROGOU A., FRESNEAU D., « Métaphore du fouragement et modèle d'exploitation collective de l'espace sans communication ni interaction pour les colonies de robots autonomes mobiles », dans Barthès J.-P., Chevrier V., Brassac C. (dir.), *6èmes Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA-98)*, Pont à Mousson, France, Afcet & Afia, novembre 1998.
- [DUR 90] DURFEE E., MONTGOMERY T., « A Hierarchical Protocol for Coordinating Multiagent Behaviors », dans *8th National Conference on Artificial Intelligence (AAAI-90)*, p. 86-93, 1990.
- [EME 89] EMERSON E.A., « Temporal and Modal Logic », dans van Leeuwen J. (dir.), *Handbook of Theoretical Computer Science*, North-Holland Publishing Company, Amsterdam, 1989.
- [FER 95] FERBER J., *Les Systèmes Multi-Agents, vers une Intelligence Collective*, InterEditions, Paris, 1995.
- [FER 97] FERBER J., « Les Systèmes Multi-Agents : un Aperçu Général », *Technique et Science Informatiques*, vol. 16, n° 8, Hermès, p. 979-1012, 1997.
- [FIN 93] FININ T., WEBER J., WIEDERHOLD G., GENESERETH M., FRITZON R., MCGUIRE J., SHAPIRO S., BECK C., « *Specification of the KQML Agent-Communication Language* », *DARPA Knowledge Sharing Initiative*, External Interface Working Group, <http://www.cs.umbc.edu/kqml/>, 1993.
- [FIP 99] FIPA, « Specification: Agent Communication Language », *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/spec/fipa99spec.htm>, septembre 1999 (Draft-2).
- [FIS 94] FISHER M., WOOLDRIDGE M., « Specifying and executing protocols for cooperative action », dans *International Working Conference on Cooperating Knowledge-Based Systems*, 1994
- [GEN 98] GENESERETH M.R., « Knowledge Interchange Format », draft proposed American National Standard (dpANS) NCITS.T2/98-004, Stanford University, CA, 1998 (<http://logic.stanford.edu/kif/dpans.htm>).
- [GRE 99] GREAVES M., HOLMBACK H., BRADSHAW J., « What is a Conversation Policy? », *Autonomous Agents'99 Special Workshop on Conversation Policies*, 1999.

- [HAD 96] HADDADI A., *Communication and Cooperation in Agent Systems: A Pragmatic Theory*, Springer-Verlag, LNCS 1056, 1996.
- [HEN 90] HENNESSY M., *The Semantics of Programming Languages: An Introduction Using Structured Operational Semantics*, Wiley, 1990.
- [HEW 77] HEWITT C., « Viewing Control Structures as Patterns of Passing Messages », *Artificial Intelligence*, vol. 8, n° 3, p. 323-364, 1977.
- [HOL 91] HOLZMANN G.J., *Design and Validation of Computer Protocols*, Prentice-Hall, 1991.
- [KIN 98] KINNY D., « The Agentis Interaction Model », dans *Fifth International Workshop on Agent Theories, Architectures and Languages (ATAL-98)*, Paris, France, p. 247-260, 1998.
- [KOL 95] KOLB M., « A Cooperation Language », dans *First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, June 1995, AAAI Press, p. 233-238, juillet 1998.
- [KON 95] KONING J.-L., DEMAZEAU Y., « Collaborative Learning Using DAI Interaction Protocols in a Telecommunication Setting », dans Barnev P. (dir.), *Proceedings of the Twentieth International Conference on Information Technologies and Programming*, Plovdiv, Bulgarie, Foi-Commerce, p. 6-16, juin 1995.
- [KON 98] KONING J.-L., FRANÇOIS G., DEMAZEAU Y., « An Approach for Designing Negotiation Protocols in a Multiagent System », dans CUENA J. (dir.), *15th IFIP World Computer Congress, IT&KNOWS Conference*, Vienne (Autriche), Budapest (Hongrie), Austrian Computer Society, Chapman and Hall, septembre 1998.
- [KRE 91] KREIFELTS T., VON MARTIAL F., « A Negotiation Framework for Antonomous Agents », dans Demazeau Y., Müller J.-P. (dir.), *Decentralized AI*, vol. II, Elsevier Science Publishers B.V., Amsterdam, p. 71-88, 1991.
- [KUW 95] KUWABARA K., ISHIDA T., OSATO N., « AgenTalk: Describing Multiagent Coordination Protocols with Inheritance », dans *Seventh International Conference on Tools with Artificial Intelligence*, Herndon, Virginia, p. 460-465, novembre 1995.
- [LAB 96] LABROU Y., *Semantics For An Agent Communication Language*, Doctoral dissertation, Univ. of Maryland, Computer Science and Electrical Engineering Dept., 1996.
- [LAB 97] LABROU Y., FININ T., A Proposal for a new KQML Specification, Technical Report TR-CS-97-03, Computer Science and Electrical Engineering Dept., Univ. of Maryland, 1997 (<http://www.cs.umbc.edu/kqml/>).
- [LAB 98] LABROU Y., FININ T., « Semantics for an agent communication language », *Agents IV, Agent Theories, Architectures and Language*, Wooldridge M., Müller J.P., Tambe M. (dir.), Springer-Verlag, LNAI, Berlin, 1998.
- [MAL 88] MALONE T.W., FIKES R.E., GRANT K.R., HOWARD M.T., « Enterprise: A Market-Like Task Scheduler for Distributed Computing Environments », dans Huberman B.A. (dir.), *The Ecology of Computation*, Elsevier Science Publishers, Amsterdam, p. 177-205, 1988.

- [MOU 99] MOULIN B., DELISLE S., CHAIB-DRAA B. (dir.), *Analyse et Simulation de Conversations – De la théorie des actes de discours aux systèmes multi-agents*, L'interdisciplinaire, Limonest, 1999.
- [NUM 90] NUMAOKA C., TOKORO M., « Conversation among Situated Agents », dans Huhns M. N. (dir.), *Tenth International Workshop on Distributed Artificial Intelligence (IWDAI-90)*, 1990.
- [OUD 00] OUDEYER P.-Y., KONING J.-L., « Modeling Soccer-Robots Strategies Through Conversation Policies », dans Mattern F., Kotz D. (dir.), *2nd IEEE International Symposium on Agent Systems and Application (ASA/MA-00)*, Springer-Verlag, Zurich, Suisse, septembre 2000.
- [PIT 99] PITT J., MAMDANI A., « Designing Agent Communication Languages for Multi-Agent Systems », *European Workshop on Modelling Autonomous Agents and Multi-Agent World (MAAMAW-99)*, p. 102-114, 1999.
- [POP 93] POPULAIRE P., DEMAZEY Y., BOISSIER O., SICHMAN J., « Description et Implémentation de Protocoles de Communication en Univers Multi-Agents », dans *1^{ères} Journées Francophones sur l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents (JFIADSMA-93)*, Afcet & Afia, Toulouse, avril 1993.
- [ROD 99] RODRIGUES-AGUILAR J.A., MARTIN F.J., NORIEGA P., GARCIA P., SIERRA C., *Towards a formal specification of complex social structures in multi-agent systems*, Springer-Verlag, LNAI, 1999.
- [SAD 91a] SADEK M.D., Attitudes mentales et interaction rationnelle: vers une théorie formelle de la communication, Thèse de doctorat, Université de Rennes I, France, 1991.
- [SAD 91b] SADEK M.D., « Dialogue acts are rational plans », dans *Proceedings of the ESCA/ETRW Workshop on the structure of multimodal dialogue*, p. 1-29, Maratea, Italie, 1991.
- [SAN 95] SANDHOLM T., LESSER V., « Issues in Automated Negotiation and Electronic Commerce: Extending the Contract Net Framework », dans *First International Conference on Multi-Agent Systems (ICMAS-95)*, San Francisco, CAM, AAAI Press, p. 328-335, juin 1995.
- [SEA 69] SEARLE J.R., *Speech acts*, Cambridge University Press, 1969.
- [SEA 85] SEARLE J.R., VANDERVEKEN D., *Foundations of illocutionary logic*, Cambridge University Press, 1985.
- [SHA 48] SHANNON C.E., « A Mathematical theory of communication », *The bell system technical journal*, vol. 27, 1948.
- [SIA 91] SIAN S. S., « Adaptation Based on Cooperative Learning in Multi-Agent Systems », dans Demazeau Y. et Müller J.-P. (dir.), *Decentralized AI, Vol. II*, Elsevier Science Publishers B.V., Amsterdam, p. 257-272, 1991.
- [SIN 92] SINGH M.P., « On the Semantics of Protocols Among Distributed Intelligent Agents », *Proceedings of the IEEE Conference on Computers and Communication*, Phoenix, AR, p. 1-14, 1992.

- [SMI 80] SMITH R.G., « The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver », *IEEE Transactions on Computers*, vol. C-29, n° 12, p. 1104-1113, 1980.
- [SMI 81] SMITH R., DAVIS R., « Framework for Cooperation in Distributed Problem Solving », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, n° 1, p. 61-70, 1981.
- [SPI 92] SPIVEY J.M., *The Z Notation*, Prentice Hall, Hemel Hempstead, 2^e édition, 1992.
- [STE 93] STEINER D., BURT A., KOLB M., LERIN C., « The Conceptual Framework of MAI2L », dans Ghedira K., Sprumont (dir.), *Fifth European Workshop on Modelling Autonomous Agents and Multi-Agent World (MAAMAW-93)*, Neuchâtel, Suisse, p. 217-230, août 1993.
- [TRO 91] TROGNON A., « L'interaction en général : sujets, groupes, cognitions, représentations sociales », *Connexions*, 57, p. 9-27, 1991.
- [TRO 92] TROGNON A., BRASSAC Ch., « L'enchaînement conversationnel », *Cahiers de Linguistique Française*, 13, p. 76-107, 1992.
- [VAN 88] VANDERVEKEN D., *Les actes de discours*, Mardaga, Bruxelles, 1988.
- [VAN 90] VANDERVEKEN D., *Meaning and speech acts*, Volume I, *Principles of language use*, Cambridge University Press, 1990.
- [VAN 91] VANDERVEKEN D., *Meaning and speech acts*, Volume II, *Formal semantics of success and satisfaction*, Cambridge University Press, 1991.
- [VAN 99] VANDERVEKEN D., « La structure logique des dialogues intelligents », dans *Analyse et Simulation de Conversations – De la théorie des actes de discours aux systèmes multi-agents*, Moulin B., Delisle S., Chaib-draa B. (dir.), L'interdisciplinaire, de Lyon, p. 61-100, 1999.
- [VON 99] VONGKASEM L., CHAIB-DRAA B., « Acl as a joint project between participants: A preliminary report », dans *First Workshop on Agent Communication Language (ACL'99)*, Stockholm, Suède, 1999.
- [WER 89] WERNER E., « Cooperating Agents: A Unified Theory of Communication and Social Structure », dans Gasser L., Huhns M. N. (dir.), *Distributed Artificial Intelligence*, vol. II, Chap. 1, p. 3-36, Morgan Kaufmann Publishers, 1989.
- [WIN 86] WINOGRAD T., FLORES F., *Understanding Computers and Cognition: A New Foundation for Design*, Addison-Wesley, 1986.

Chapitre 4

Modèles de coordination d'agents cognitifs

4.1. Problématique de la coordination

Les systèmes multi-agents (SMA) proposent une approche originale de conception de systèmes intelligents et coopératifs. Ils se caractérisent par la présence d'agents (par exemple : cognitifs, intentionnels, rationnels, etc.) plus ou moins autonomes et hétérogènes évoluant dans un environnement partagé et dynamique. La conception de tels systèmes induit de nouveaux problèmes inhérents à la vision partielle des agents, la poursuite d'objectifs locaux et l'entrelacement des activités des agents. La coopération entre agents, en vue d'engendrer un comportement global cohérent du SMA, requiert des mécanismes élaborés de coordination afin d'éviter des conflits potentiels et de favoriser la synergie des activités des agents [BON 88].

La coordination multi-agent fait souvent référence au processus qui contrôle la prise de décision et guide le comportement global inhérent à l'exécution d'une collection d'agents. Elle peut être définie comme un problème *de prise de décision distribuée sous incertitude*, comme un ensemble de *questions relatives à la connaissance et l'action*, ou encore comme un *problème de conception* telle que la coordination de *frameworks* pour des organisations homme-machine [GAS 92].

Le processus de coordination implique à un instant donné un ensemble d'agents du SMA et les aide à atteindre un état global cohérent c'est-à-dire où les états locaux des différents agents sont compatibles en termes d'objectifs, d'actions, d'utilisation de ressources, etc. Ainsi, tout au long de ce processus, les agents vont œuvrer pour

éviter des situations conflictuelles (antagonisme, interblocage, famine, etc.) pouvant être engendrées par une mauvaise utilisation des ressources partagées, par l'existence de connaissances et de visions incompatibles, etc. Au-delà de la résolution de conflits potentiels, la coordination présente un intérêt majeur, celui de favoriser la synergie des agents en leur permettant de bénéficier de leurs capacités respectives, et de tirer profit des actions des uns et des autres. On parlera de situations d'interaction favorables (ou positives) entre agents.

La coordination reste donc une des problématiques majeures dans le domaine des SMA. Plusieurs solutions ont été explorées et des techniques très variées ont été proposées. Ces techniques dépendent des définitions adoptées et des perspectives à travers lesquelles la coordination est abordée. Notons toutefois que, dans ce chapitre, nous nous intéressons à la coordination comme un processus *conscient* élaboré par des agents disposant de réelles capacités de raisonnement, tels que les agents cognitifs et/ou rationnels. Ce type de coordination n'inclut pas *la coordination réactive* qui repose essentiellement sur le mécanisme de perception/réaction et suppose que le contexte est très évolutif rendant ainsi l'anticipation sur la suite des actions difficilement réalisable [FER 95]. Cet aspect de la coordination s'applique aux systèmes réactifs qui ne sont pas abordés dans ce chapitre.

Ce chapitre est organisé comme suit : la section 4.2 rappelle quelques définitions de la coordination parmi les plus consensuelles à travers les différentes perspectives de la coordination. La section 4.3 présente un scénario fondé sur la métaphore des agents déménageurs inspiré de l'exemple initialement traité dans [MOR 96]. Ce scénario, tout en étant simple, est suffisamment riche pour illustrer différentes situations de coordination. La section 4.4 définit les dimensions d'un mécanisme de coordination. La section 4.5 dresse un panorama des approches les plus utilisées dans le domaine de la coordination. La section 4.6 est dédiée à la coordination d'actions par planification qui représente une approche centrale de la coordination des SMA. La section 4.7 établit les critères d'évaluation des modèles de coordination en vue d'une étude comparative des modèles présentés dans les sections 4.5 et 4.6. La section 4.8 conclut ce chapitre en soulignant les problèmes encore ouverts en coordination et les axes de recherche correspondants.

4.2. Perspectives de la coordination

4.2.1. *Pluralisme*

Une des premières motivations de la coordination a été l'incorporation de points de vue conflictuels lors de la résolution distribuée de problèmes. Ce concept a été d'abord introduit par Kornfeld et Hewitt [KOR 81] dans leur travail *Métaphore de la communauté scientifique* où la concurrence et la divergence de points de vue sont

les clés de son succès. Ils ont développé un langage appelé ETHER qui permet un grand degré de parallélisme de systèmes de résolution de problèmes. Un peu plus tard, Carl Hewitt [HEW 91] a proposé un cadre conceptuel pour raisonner sur des processus ayant des points de vue conflictuels. Pour Bond et Lesser [BON 88], la coordination permet d'« *aligner des activités de différentes perspectives* ». Elle apporte une solution aux problèmes suivants :

1) Comment garantir que les agents agissent de manière cohérente lors d'une prise de décision ou du choix des actions futures, ainsi que lors de la recherche d'un compromis entre les effets locaux et globaux des décisions afin d'éviter les interactions nuisibles ou négatives ?

2) Comment permettre aux agents de représenter et de raisonner sur les actions, les plans et les connaissances des autres agents pour se coordonner avec eux ? Comment raisonner sur l'état du processus de coordination (initialisation et achèvement du processus) ?

3) Comment reconnaître et réconcilier des points de vues divergents et des intentions conflictuelles à travers un ensemble d'agents pour coordonner leurs actions ?

4.2.2. Recherche de la cohérence globale du système

Une deuxième motivation pour les travaux en coordination a été la recherche de la cohérence globale du système en tant qu'entité en dépit de l'absence d'un contrôle centralisé ou d'un point de vue global. Pour Les Gasser [GAS 92], la coordination est un problème de contrôle distribué où il s'agit de contrôler des décisions de contrôle ou encore des décisions relatives au choix des actions futures lors de la résolution d'un problème. Le degré de coordination dépend de la qualité du contrôle de décision qui implique la connaissance des alternatives, les critères de décision et les procédures applicables aux critères et aux alternatives.

Par connaissances de contrôle, on entend toutes celles capables d'informer et d'influencer les décisions de contrôle. La complexité d'un système de coordination dans ce cas est fonction du degré de décisions de contrôle à prendre : leur nombre, leur simultanéité, (a)synchronisme, (dé)centralisation, etc.

Plusieurs variantes des systèmes de contrôle en fonction de ces dimensions ont été définies. On dira que deux systèmes de contrôles sont équivalents par rapport à leurs effets s'ils mènent à la même séquence de choix d'actions.

Pour Katia Sycara [SYC 91], la coordination peut être définie comme la gestion des interdépendances entre les activités des agents ou, de manière plus pragmatique, comme le choix, l'ordonnancement et l'affectation des actions au cours du temps

pour essayer de maximiser l'ensemble des critères de décision qui peut varier dans le temps. La coordination implique non seulement les interactions localisées des agents lors de la résolution d'un problème spécifique, mais aussi et à long terme les organisations d'agents qui peuvent supporter l'activité de la résolution du problème actuel ou futur. Ainsi, trois classes de modèles de coordination peuvent être distinguées :

- les modèles des actions futures des autres agents, les ordonnancements et/ou plans ;
- les modèles des objectifs, des désirs et des buts des autres agents ;
- et les modèles des capacités des autres agents.

Plusieurs travaux se sont également intéressés à la coordination dans le domaine de la prise de décision distribuée ou encore la théorie des jeux [ROS 94, ZLO 91]. La définition proposée par Sascha Ossowski [OSS 99] est intéressante pour sa généralité. Elle distingue deux cas : la coordination centralisée et la coordination distribuée.

La *coordination centralisée* repose sur l'existence d'un agent central ou coordinateur. La résolution de la coordination peut se ramener à la résolution d'un problème d'optimisation ou à celle d'un problème de satisfaction de contraintes.

Dans le cas où un problème de coordination est exprimé sous la forme d'un problème d'optimisation, il peut être défini comme un triplet (V, D, U) où :

- $V = \{v_1, \dots, v_n\}$ est l'ensemble fini de variables de décision représentant les objets de la coordination ;
- $D = \{D_1, \dots, D_n\}$ est l'ensemble des domaines respectifs des variables $v_1, \dots, v_n$;
- $U: X \rightarrow \mathbb{R}$ est une fonction d'utilité où $X = D_1 \times \dots \times D_n$ désigne l'ensemble des instanciations possibles représentant l'espace des décisions, c'est-à-dire l'ensemble de toutes les combinaisons possibles des objets de coordination.

Formulé ainsi, résoudre un problème de coordination revient à trouver une *instanciation optimale* x , c'est-à-dire $\forall y \in X, U(y) \leq U(x)$

Dans le cas où un problème de coordination est exprimé sous-forme d'un problème de satisfaction de contraintes, la fonction d'utilité est substituée par un ensemble de contraintes représentant les dépendances entre les objets de la coordination. Le problème de la coordination est ainsi défini comme un triplet (V, D, C) où V et D sont définis comme dans le modèle précédent (*i.e.* résolution d'un problème d'optimisation), alors que $C = \{C_i\}$ représente l'ensemble des contraintes de dépendance. Chaque contrainte C_i peut être considérée comme un sous-ensemble de $X = D_1 \times \dots \times D_n$.

Exprimé ainsi, résoudre un problème de coordination consiste à trouver une *instanciation consistante* x , qui satisfait toutes les contraintes de dépendances.

La *coordination distribuée* est particulièrement adaptée dans le cas d'agents rationnels possédant éventuellement des intérêts conflictuels, c'est-à-dire que chaque agent développe une stratégie propre pour maximiser sa fonction d'utilité. La coordination cherche à trouver un équilibre ou un compromis entre les intérêts des agents impliqués dans le processus de coordination. Cet aspect sera détaillé plus loin (voir paragraphe 4.5.4). Notons cependant qu'une telle coordination requiert souvent un mécanisme de négociation ou de médiation entre agents [LAN 93, MÜL 96, ZLO 96].

4.3. Scénario des agents déménageurs

Dans cette section, nous introduisons le scénario des agents déménageurs qui sera utilisé pour illustrer les concepts de coordination présentés dans ce chapitre. Nous commençons par une description intuitive et informelle du scénario. Il s'agit d'un ensemble d'agents dont le but est d'assurer le déménagement d'une maison (tâche T) constituée de plusieurs pièces, à vider de leurs meubles. Le propriétaire de la maison fait appel à une agence de déménagement. Le gestionnaire de l'agence délègue cette tâche à un ensemble d'agents déménageurs ($A_1, A_2, \dots, A_k$). Nous distinguerons plusieurs cas de délégation (ou d'allocation) de la tâche.

Du point de vue de l'agent gestionnaire :

- 1) il délègue aux agents A_j la tâche T sans décomposition préalable ;
- 2) il décompose la tâche T en sous-tâches ($T_1, T_2, \dots, T_n$) et alloue les sous-tâches T_i aux agents A_j ;
- 3) il planifie la tâche T et délègue le plan global aux agents A_j ;
- 4) il planifie la tâche T , répartit le plan global en sous-plans qu'il alloue aux agents A_j ;
- 5) il délègue un sous-plan, une sous-tâche, etc., à un groupe d'agents dans le cas où certaines actions (ou plans) nécessiteraient plus d'un agent à la fois.

Du point de vue de l'agent déménageur, deux situations sont possibles.

Situation de coopération : les agents déménageurs n'ont pas d'intérêts individuels, mais partagent tous le même but qui est de réaliser la tâche T attribuée par le gestionnaire de l'agence. Le degré de coopération entre ces agents dépendra de la nature des sous-tâches à effectuer. La coordination entre agents déménageurs portera sur la décomposition de la tâche T (cas 1), sur les plans individuels (cas 2 et

4), les actions collectives (cas 3) ou jointes (cas 5), etc. Plusieurs cas peuvent être déclinés. On peut citer à titre d'exemple :

- le cas d'un objet léger : deux agents ne doivent pas soulever le même objet (actions redondantes), ne doivent pas décider de soulever ce même objet (décisions incompatibles, intentions conflictuelles, etc.) ;
- le cas d'un objet lourd : un agent ne doit pas tenter de soulever seul cet objet (besoin de coopération, de synchronisation d'actions), ne doit pas décider de soulever seul cet objet (coordination des décisions, des intentions, des plans, etc.) ;
- le cas d'un couloir étroit : deux agents ne doivent pas s'engager *simultanément* dans ce couloir (ordonnancement des actions), etc.

Situation de compétition : les agents déménageurs ont des intérêts individuels. Par exemple, chaque objet déménagé rapporte une rétribution à l'agent qui s'en est chargé. Dans ce cas, les agents risquent d'entrer en compétition pour porter des objets légers. Par contre, pour des objets lourds, ils seront *amenés à coopérer* en formant des coalitions par exemple.

4.4. Dimensions de la coordination

Appréhender un modèle de coordination dans sa globalité nécessite d'intervenir selon trois dimensions fortement corrélées :

- 1) la structure de l'agent qui détermine le sujet de la coordination : *que coordonner ?*
- 2) le modèle de spécification du problème de coordination : *à travers quelle représentation ?*
- 3) les techniques et mécanismes de coordination : *comment ?*

4.4.1. Structure d'un agent

Situé dans un environnement, un agent est amené à interagir avec d'autres agents et à réguler son comportement en conséquence. Le comportement d'un agent dépend de sa structure interne (croyances, intentions, connaissances, etc.) et de sa perception externe au sens large (perception, communication, interaction).

La structure interne d'un agent est un ensemble de composants internes généralement inaccessibles aux autres agents (voir figure 4.1). Elle définit son architecture qui dépend du type d'agent considéré (voir chapitre 2). Ainsi, on trouvera le concept des états mentaux ou attitudes intentionnelles [BRA 87, RAO 91] dans le cas d'agents intentionnels, les connaissances et les mécanismes de

raisonnement, de délibération et de prise de décision dans le cas des agents cognitifs ainsi que des engagements dans le cas d'agents sociaux [CAS 98].

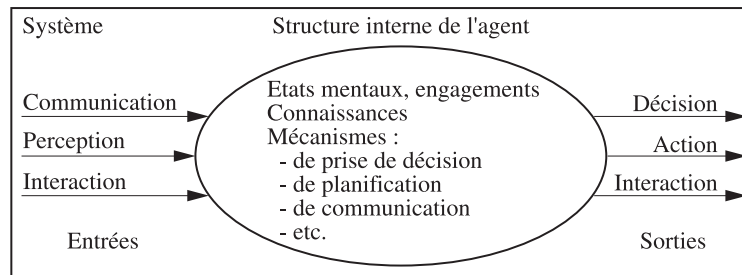


Figure 4.1. Paramètres de la coordination

Pour un observateur externe du SMA, un agent peut être perçu comme une entité qui subit l'influence de son environnement et réagit en conséquence. En effet, l'agent dispose (outre ses connaissances propres, telles que son expertise d'un domaine donné) de connaissances sur son environnement incluant les autres agents. Ces connaissances sont dynamiquement mises à jour en fonction de la perception qu'il a de son environnement et des communications ou interactions qu'il opère avec les autres agents. Le comportement d'un agent est donc perceptible en termes d'actions, de plans, de décisions ou de buts déclarés, d'interactions, etc., qui découlent de sa structure et ses mécanismes internes.

Coordonner le comportement d'un agent avec ceux d'autres agents revient donc à agir sur l'une de ses composantes internes pour influencer ses actions, ses plans, ses décisions, etc. Ce qui répond à la question : « Que faut-il coordonner ? »

Dans le scénario des agents déménageurs, chaque agent peut être amené à modifier ses intentions qui donnent lieu à des actions (par exemple : soulever un objet, le porter, le charger, s'engager dans un couloir étroit en portant un objet encombrant, etc.) ; ou encore à ses décisions et plans (décider de s'engager dans le couloir, synchroniser son plan avec celui d'un autre, etc.).

4.4.2. Modèle de spécification du problème de coordination

La spécification d'un modèle de coordination nécessite la caractérisation des situations de coordination. Il s'agit de représenter les conditions d'une situation qui requiert la coordination, par exemple une situation critique (deux actions qui font appel à une même ressource à un seul point d'accès). Après la caractérisation des

situations de coordination positives ou négatives, l'agent doit être en mesure de les détecter automatiquement et de déterminer le *sujet* de la coordination.

Le modèle de spécification du problème de coordination répond à la question : « *A travers quelle représentation ?* » Cette représentation dépend du sujet de la coordination. Par exemple, s'il s'agit d'une coordination de plans, le modèle de spécification sera fonction du formalisme de représentation de plans. Si, par contre, les agents doivent coordonner leurs décisions, le modèle de spécification dépendra de la représentation des décisions, des buts, etc.

Dans la littérature consacrée aux SMA, les modèles utilisés sont souvent issus de l'intelligence artificielle ou des modèles des systèmes concurrents. Ainsi, sont utilisés des modèles basés sur des formalismes logiques (logique des prédicats, logique modale, logique déontique, logique temporelle et leurs variantes, etc.), la théorie des graphes de buts et/ou l'algèbre des processus, des formalismes pour la concurrence tels que les réseaux de Petri, etc.

EXEMPLE.— Modélisation du monde des déménageurs

Initialement, le séjour contient 2 fauteuils, une table, une lampe, une partition de musique, un tapis :

Position(Tapis, Séjour), Position(Table, Séjour), Position(Lampe, Séjour),
Position(Fauteuil_1, Séjour), Position(Fauteuil_2, Séjour),
Sur (Lampe, Table), Sur (Partition, Table), Sur(Table, Tapis),
Sur (Fauteuil_1, Sol), Sur(Fauteuil_2, Sol), etc.

L'environnement : l'environnement est représenté par un ensemble de prédicats : $P = \{p, \neg p, q, \neg q, \dots\}$.

Un état de l'environnement est un sous-ensemble S consistant de P (c'est-à-dire si $p \in S$ alors $\neg p \notin S$). Par exemple : l'état initial du scénario.

Extension de la négation à un ensemble de prédicats. Soit :

$$S \in 2^P, S \subset P; \neg S = \{\neg \xi / \xi \in S\}$$

Un état partiel de l'environnement est défini par : $\forall \xi \in P, |\{\xi, \neg \xi\} \cap S| \leq 1$

Un état total de l'environnement est défini par : $\forall \xi \in P, |\{\xi, \neg \xi\} \cap S| = 1$

Une action α est définie par :

- un label (nom de l'action),
- un identificateur (identité de l'agent qui effectue l'action),

- $\text{Pré}(\alpha)$: ensemble de préconditions qui doivent être vraies pour que α soit exécutable,
- $\text{Post}(\alpha)$: un ensemble de postconditions qui seront vraies après l'exécution de α .

EXEMPLE.— Soit l'action $\alpha = \text{Id.Déplacer}(X)$

$\text{Pré}(\text{Id.Déplacer}(X)) :$

$\text{Libre}(X)$

$\text{Sur}(X, T)$

$\text{Capacité_Agent}(\text{Id}, C) \wedge C > \text{Poids}(X)$

$\text{Position}(\text{Id}, Y) \wedge \text{Position}(X, Z) \wedge Y=Z$

$\text{Post}(\text{Id.Déplacer}(X)) :$

$\text{Sur}(X, \text{Sol})$

$\text{Libre}(T)$

$\text{Pré}(\alpha)$ et $\text{Post}(\alpha)$ sont des états partiels de l'environnement. L'exécution de l'action α à partir d'un état S , produit un état S' tel que :

$$S \xrightarrow{\alpha} S' \Leftrightarrow \text{Pré}(\alpha) \subset S \text{ et } S' = (S \setminus \neg \text{Post}(\alpha)) \cup \text{Post}(\alpha)$$

Soit $\text{Act} = \{a_1, a_2, \dots, a_n\}$ l'ensemble des actions possibles.

EXEMPLE.— $\text{Id.Déplacer}(\text{Objet}) ; \quad \text{Id.Soulever}(\text{Objet}) ; \quad \text{Id.Transporter}(\text{Objet}) ;$
 $\text{Id.Charger}(\text{Objet}), \text{ etc.}$

Soit $\Re$ une relation d'ordre partiel définie sur Act .

EXEMPLE.— Supposons que les deux actions $\text{Déplacer}(Z)$ et $\text{Déplacer}(Y)$ sont parallélisables (voir figure 4.2).

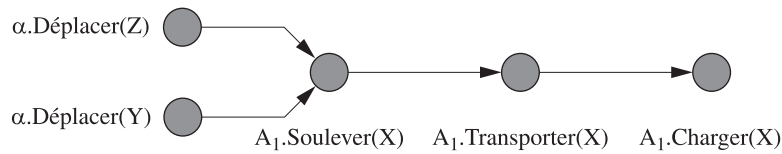


Figure 4.2. *Ordre partiel des actions du plan*

Un plan Π est un ordre partiel $\Re_\Pi$ défini sur un sous-ensemble $A_\Pi \subset \text{Act}$ tel que $\Pi = \{ A_\Pi, \Re_\Pi \}$ est un graphe acyclique où A_Π est l'ensemble des nœuds et $\Re_\Pi$ l'ensemble des arcs tels que $\Re_\Pi \subset A_\Pi \times A_\Pi$.

Dans la figure 4.2, les actions (représentées par des nœuds, par exemple : $\text{Charger}(X)$) et les arcs définissent l'ordre partiel entre ces actions.

Un but est un état partiel qui contient l'état but à atteindre. Dans la figure 4.3, le but de A_1 est $Position(Camion, X)$ qui est une postcondition de l'action $Charger(X)$. Dans le plan de A_1 , il est supposé qu'il existe un agent α (à déterminer) qui peut déplacer l'objet Z (instancié par partition) et l'objet Y (instancié par lampe) se trouvant sur la table afin que A_1 puisse à son tour soulever la table, la transporter et la charger dans le camion (voir figure 4.3).

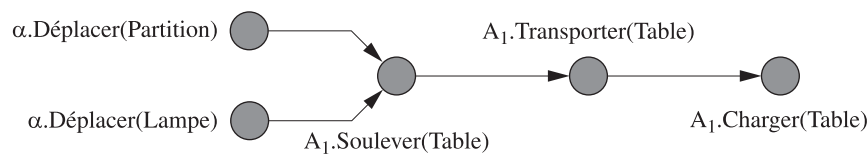


Figure 4.3. Un plan partiellement instancié

4.4.3. Mécanismes de coordination : modèles orientés tâches versus modèles orientés agents

Les mécanismes de coordination correspondent aux techniques et moyens mis en œuvre pour coordonner les comportements d'agents à différents niveaux d'abstraction. Ils tentent de répondre à la question : « Comment coordonner ? » et dépendent du modèle de spécification adopté pour un sujet de coordination donné. On distingue deux grandes classes de modèles de coordination : les modèles orientés tâches et ceux orientés agents.

4.4.3.1. Modèles de coordination orientés tâches

Il s'agit de modèles applicables dans un contexte de résolution distribuée de problèmes (*Distributed Problem Solving*). Dans ces modèles de coordination, on présuppose l'existence d'un but global partagé par un groupe d'agents au sein d'une organisation. C'est le cas par exemple de nos agents déménageurs dans une situation coopérative où le gestionnaire a déjà distribué la tâche en sous-tâches et alloué chacune d'elles à un agent ou un groupe d'agents :

- le but global des agents est d'accomplir la tâche T de déménagement allouée par le gestionnaire de l'agence ;
- le but individuel de chaque agent A_i est d'accomplir une sous-tâche T_j selon l'allocation du gestionnaire (par exemple : déménager une pièce).

La coordination est nécessaire pour optimiser l'efficacité globale du groupe d'agents et non pas d'un point de vue individuel des agents. Les agents sont typiquement dans une situation de coopération. Des interactions apparaissent au fur et à mesure que les agents essaient de réaliser leur tâche individuelle, par exemple :

s'entraider pour porter un objet lourd, ou se synchroniser pour ne pas emprunter un couloir étroit simultanément, etc.

4.4.3.2. *Modèles de coordination orientés agents*

Il s'agit de modèles applicables dans le contexte d'agents (semi-)autonomes ne partageant pas *forcément* un but global sinon la cohérence et la viabilité du système. Dans ce cas de figure, les agents en tant qu'entités individuelles cherchent à optimiser leurs propres objectifs (par exemple : gagner le maximum dans la situation de compétition des agents déménageurs). Même en situation de compétition, la coordination est *nécessaire* pour accomplir des tâches qu'un agent ne peut réaliser seul ce qui lui permet d'augmenter ses gains.

A nouveau, dans le cas de modèles orientés agents, on peut distinguer deux grandes tendances : la coordination d'agents pris individuellement (par exemple : coordonner les plans individuels des agents déménageurs A_1 et A_2) et la coordination d'un groupe d'agents (par exemple : A_1 et A_2 élaborent un seul plan appelé Plan_Joint [GRO 96, HUB 96, KIN 92], Agency, Teamwork, etc.) qui vont mener des actions jointes à partir de connaissances et/ou croyances collectives (connaissances communes, attitudes intentionnelles jointes, croyances mutuelles, activités de groupe, etc.).

Les deux classes de modèles de coordination orientés tâches et orientés agents reposent sur les mécanismes détaillés dans les approches présentées dans les sections 4.5 et 4.6, et éventuellement les combinent.

4.4.3.3. *Mécanisme de coordination*

Indépendamment du modèle de coordination considéré, et relativement à un sujet de coordination, un mécanisme de coordination peut être résumé en cinq phases (voir figure 4.4) n'excluant pas des traitements parallèles relatifs à plusieurs sujets de coordinations si les agents en sont capables.

4.4.3.4. *Exemple de coordination de plans*

Soient deux agents déménageurs A_1 et A_2 qui doivent planifier à partir de l'état partiel courant : Sur (Lampe, Table), Sur(Table, Tapis), Position(Tapis, Séjour), Position (Camion, Rue), etc. ; pour atteindre leurs buts respectifs But₁ (Position (Table, Camion)) et But₂ (Position (Tapis, Camion)), A_1 et A_2 forment respectivement les plans suivants (voir figure 4.5).

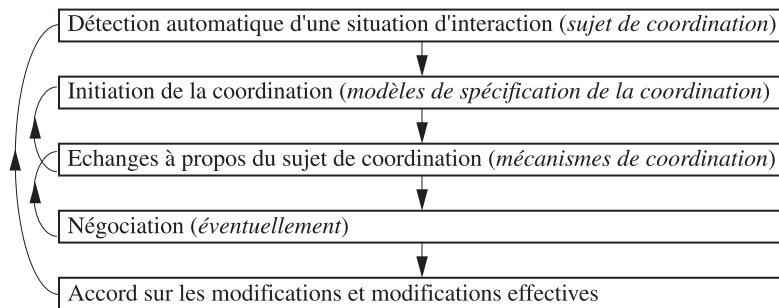


Figure 4.4. Cycle de base d'un mécanisme de coordination

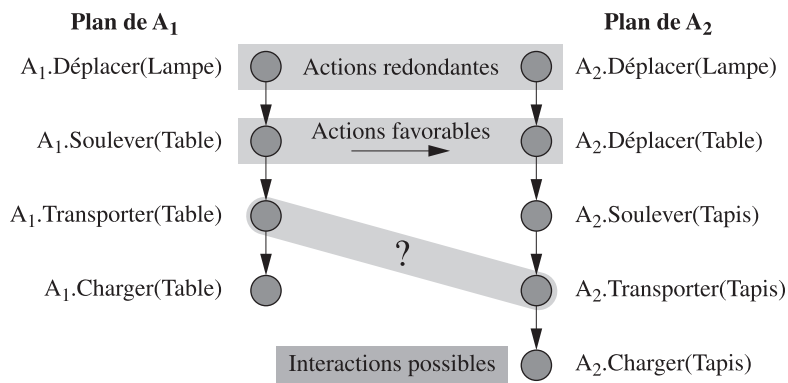


Figure 4.5. Plans initiaux non coordonnés

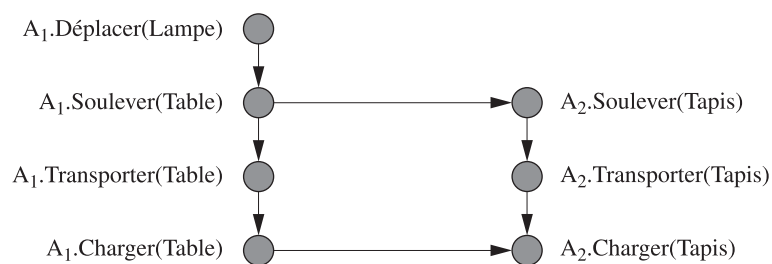


Figure 4.6. Plans coordonnés de A_1 et A_2

Après coordination, un plan coordonné possible pour A_1 et A_2 est tel que décrit en figure 4.6.

4.5. Approches de coordination d'agents

Cette section présente quelques approches des plus significatives dans le domaine de la coordination SMA aussi bien celles orientées tâches que celles orientées agents. La section 4.6 est entièrement consacrée aux travaux concernant la coordination par planification qui constitue une approche majeure de la coordination SMA.

4.5.1. Coordination basée sur la résolution distribuée de problèmes

Un des premiers modèles de coordination orientés tâches a été proposé par Lesser dans [LES 81]. Ce modèle appelé FA/C (*Functionally Accurate Model*) (modèle fonctionnel exact) peut être caractérisé par la distribution du contrôle, et de ce fait, il a constitué une approche novatrice au début des travaux sur la coordination des SMA. Son intérêt réside dans le fait que les agents n'ont pas besoin d'avoir localement toutes les informations pour résoudre leur problème, mais doivent interagir pour acquérir dynamiquement les informations requises et se coordonner en fonction de leurs résultats partiels. Il s'agit de définir des stratégies de coordination basées sur un mécanisme de contrôle distribué qui suppose une interdépendance entre les activités des agents. Ces derniers échangent des informations de haut niveau (dites *méta*) tels que leurs choix, leurs préférences, leurs priorités, leurs buts, etc.

La critique majeure adressée à ce modèle est que le niveau *méta* est défini de manière statique, ce qui rend l'adaptation du mécanisme de coordination difficile.

Depuis ce modèle, plusieurs tentatives d'amélioration du mécanisme de distribution du contrôle ont été proposées [LES 91].

Le modèle proposé par E. H. Durfee et V. R. Lesser [DUR 91] appelé planification globale partielle (PGP : *Partial Global Planning*) se caractérise par sa flexibilité et sa dynamique. Contrairement au modèle précédent (FA/C), ici le niveau *méta* est dynamique. Dans ce modèle, les agents coordonnent dynamiquement leurs plans en échangeant des informations pertinentes pour l'exécution de ces plans. Les informations pertinentes sont définies à trois niveaux de représentation des plans ; et chaque agent gère les trois niveaux comme suit :

- plan local : chaque agent développe un plan local détaillé dont il contrôle l'exécution ;

- plan nodal : ce plan est une abstraction du plan local qui inclut des informations pertinentes destinées à être communiquées aux autres agents ;
- plan global partiel : ce plan contient les buts des autres agents, dans ce sens il est global. Par contre, il est partiel car il ne concerne qu'un sous-ensemble d'agents restreint à ceux susceptibles d'avoir des interactions avec l'agent propriétaire du plan.

Les interactions concernent la communication des plans et des buts à un niveau d'abstraction approprié. Les agents récepteurs infèrent les buts et les actions des agents émetteurs et décident des actions à planifier en leur faveur. Ainsi, cet échange de plans ou buts améliore les capacités de prédiction d'un agent relativement aux plans des autres agents en vue d'une coordination plus efficace.

Ce modèle a été appliqué au problème de la surveillance de la circulation de véhicules dans un espace divisé en zones (DVMT : *Distributed Vehicle Monitoring Testbed*). Chaque agent est responsable d'une zone et les interactions en terme de surveillance portent sur les recouvrements de ces zones. On remarquera ici que la distribution du problème porte sur l'espace à surveiller et que les tâches allouées aux agents sont similaires. Dans ce cas particulier, les interactions sont limitées aux agents responsables des régions voisines selon une topologie spatiale.

Decker a ensuite généralisé ce modèle [DEC 92] en proposant le modèle GPGP (*Generalized Partial Global Planning*), où la coordination doit respecter des contraintes d'exécution temps réel. Par conséquent, la structure des buts a été augmentée et affinée à travers un arbre de buts étendu organisant les buts de manière hiérarchique et incluant diverses informations telles que : l'ordre des buts, leurs durées, leurs priorités, les ressources qui leur sont nécessaires, le plan solution en cours, les contraintes imposées aux solutions attendues, etc.

Dans cette généralisation (GPGP), le mécanisme de coordination se base sur trois relations de dépendance :

- les relations de coordination entre actions : inhiber, annuler, causer, faciliter, contraindre, etc. ;
- les relations entre les graphes de buts : recouvrement, extension, suffisance, concurrence, etc. ;
- les relations temporelles : dures (*hard*), négociables, légères (*soft*) pour représenter des préférences locales des agents.

Ainsi, après avoir construit un plan local des activités à moyen terme, et après avoir déterminé les relations de dépendance entre ces activités, chaque agent communique avec ses voisins et met à jour sa représentation individuelle à partir des échanges. L'intérêt principal de ce modèle est la possibilité de *prédire* les sujets de

coordination et de les traiter en temps utile voire d'anticiper les traitements [DEC 93, DEC 95].

4.5.2. Coordination basée sur les structures organisationnelles

Les structures organisationnelles jouent un rôle important pour définir et guider les rôles des agents et par conséquent leurs comportements au sein d'une organisation. De manière externe, une organisation peut être vue comme une délimitation des responsabilités et des actions des agents. En contrepartie, la structure organisationnelle peut constituer un moyen de pallier les limitations des agents en termes de compétences ou d'aptitudes à réaliser des tâches complexes.

Plusieurs types d'organisation existent dans la littérature consacrée aux SMA. Une caractéristique commune aux organisations est que l'on peut les voir comme des distributions (statiques ou dynamiques) selon trois dimensions pouvant être combinées :

- la distribution fonctionnelle : elle consiste à répartir les spécialisations, les rôles ou les fonctions des agents au sein d'une organisation, et ce en tenant compte de leurs aptitudes, de la nature des tâches qui leur incombent et des interactions possibles entre eux (par exemple, une spécialisation des agents déménageurs par type d'objet à déménager : porteurs d'objets lourds, porteurs d'objets fragiles, etc., ou encore une hiérarchie des agents : gestionnaire et agents déménageurs sous ses ordres, etc.) ;
- la distribution spatiale : une telle distribution concerne l'affectation des agents à des régions bien délimitées suite au découpage de l'environnement dans lequel ils évoluent (par exemple : affectation des agents déménageurs aux pièces de la maison). Cette répartition de l'espace n'est pas nécessairement une partition car les recouvrements peuvent être introduits pour améliorer la robustesse des traitements, notamment dans le cas des transitions d'une région à une autre voisine (par exemple : dans l'exemple DVMT, les agents ont des rôles identiques mais sont affectés à des régions différentes de l'espace qui se recouvrent) ;
- la distribution temporelle : ce type de distribution relève des organisations dynamiques. C'est le cas des structures où les agents peuvent évoluer, apprendre, acquérir de nouvelles connaissances. Par conséquent, les processus inhérents à leurs exécutions, leurs communications ainsi que leurs interactions peuvent changer et influencer la structure organisationnelle elle-même (par exemple : coalitions ponctuelles des agents déménageurs pour porter des objets lourds).

Du point de vue de la coordination, et à un plus haut niveau d'abstraction, une organisation peut être vue [GAS 84] comme un mécanisme de coordination qui se situe au niveau global, c'est-à-dire au niveau du réseau d'agents.

4.5.2.1. Organisations statiques

Lorsqu'elle est définie statiquement, une structure organisationnelle fige *a priori*, lors de la conception du SMA, les comportements des agents qui sont à un certain niveau d'abstraction coordonnés et réglementés par la structure qui les réunit. Dans ce type de structure, on retrouve les techniques de *coordination par réglementation* qui consistent à introduire des règles de comportement que les agents doivent respecter afin d'éviter des conflits potentiels. De ce fait, plusieurs conflits sont résolus *a priori*. Des exemples types de la coordination par réglementation sont : le code de la route, les règles de déplacement dans un parc pour un ensemble de robots autonomes et mobiles. D'autres approches voisines se fondent également sur les normes (agents normatifs) [DIG 99], les dépendances et conventions sociales (agents sociaux) [CAS 98, CAS 99, SIC 93, SIC 94], etc.

L'intérêt généralement escompté de l'emploi de telles techniques est la réduction des coûts de communication entre agents [BRI 95] et l'élimination de conflits potentiels dès la conception. En revanche, les lois pèsent sur l'*autonomie* des agents dont il faut *délimiter* le comportement.

Dans [GAS 84], le concept d'organisation est défini comme « *une coordination à long terme* ». Cette définition peut être affinée en distinguant une organisation statique d'une organisation dynamique. Dans le premier cas, l'organisation statique peut être en effet vue comme un moyen de coordination à long terme, alors que dans le second cas, et d'un point de vue purement temporel, elle constitue un moyen de coordination plutôt à court terme puisque les fondements même de l'organisation peuvent changer au cours du temps (par exemple : formation et dissolution de coalitions).

4.5.2.2. Organisations dynamiques

Un problème de coordination dans le cas d'une organisation dynamique peut être formulé selon trois niveaux [SYC 89] :

- 1) au niveau de la conception de l'organisation : comment organiser et répartir les tâches dynamiquement pour atteindre les objectifs du système pouvant eux-mêmes varier dans le temps ?
- 2) au niveau de la dynamique de l'organisation : comment sauvegarder les compétences et les connaissances du système quand l'organisation change ?
- 3) au niveau cognitif : comment représenter les connaissances organisationnelles et quel niveau de partage (connaissances, contrôle, etc.) est requis entre agents ?

Différentes approches pour la dynamique des organisations ont été explorées. On peut citer à titre d'exemple les agents flexibles définis par Toru Ishida [ISH 92], qui

peuvent se restructurer en fonction des évolutions environnantes (autres agents, environnement, etc.).

Un autre domaine très significatif pour les organisations dynamiques est celui de la formation de coalitions dans les SMA (voir paragraphe 4.5.5). Par définition, les coalitions sont des regroupements ponctuels d'agents compétitifs (auto-intéressés). L'intérêt des coalitions est qu'elles se forment et se dissolvent en fonction des entrées du système (par exemple : les buts nouvellement arrivés, les agents qui intègrent ou quittent le système, etc.) garantissant ainsi une meilleure adaptation aux changements de l'environnement.

4.5.3. Coordination basée sur les protocoles de coopération

A l'origine des langages d'interaction se situent les langages de coordination proposés par Gelernter et Carriero [GEL 92]. Ces auteurs les explicitent ainsi : « *Une computation doit communiquer (interagir) avec son environnement ou ne servira à rien. L'environnement peut inclure des humains, d'autres processus, etc. La forme de cette interaction est du domaine de la coordination et le point de mire des langages de coordination* ».

Une approche de plus en plus développée repose sur les protocoles de coordination, appelés également protocoles d'interaction. Dans ce cas, la coordination est basée sur la communication entre agents, celle-ci respecte un protocole défini et agréé par les agents. Ces derniers se coordonnent en émettant des requêtes aux autres agents, en leur envoyant des informations, en leur déléguant des tâches, etc.

Généralement, un protocole de coopération décompose une tâche en sous-tâches et les distribue, suivant une stratégie donnée, sur des agents du système. Cette stratégie doit respecter les critères suivants [DUR 87] :

- éviter le recouvrement en ressources critiques ;
- allouer les tâches aux agents en fonction de leurs capacités ;
- tenir informés les agents des différentes allocations de tâches ;
- allouer les responsabilités aux agents de manière couvrante pour assurer la cohérence globale ;
- allouer les tâches interdépendantes aux agents en optimisant leur distance spatiale pour minimiser les coûts de communication ;
- réallouer les tâches s'il est nécessaire de les accomplir de manière urgente.

Un des premiers protocoles de coopération a été proposé par D. Smith. Il est connu sous le nom du protocole du réseau contractuel (*Contract Net protocol*)

[SMI 80] et respectant les critères énoncés ci-dessus. Plus récemment, plusieurs protocoles de coopération ont été proposés dans le cadre des langages d'interaction d'agents tels que FIPA-ACL (ACL : *Agent Communication Language*). D'autres protocoles et plusieurs de leurs variantes ont été proposés dans la littérature SMA en vue de rendre opérationnels des protocoles pour les enchères, les appels d'offre, etc.

4.5.4. *Modèles de négociation et de prise de décision distribuée*

La négociation dans les SMA constitue une étape importante dans un processus de coordination [DAV 83]. La négociation est un processus dynamique qui vise à établir des accords entre agents et à induire des engagements de leur part à court ou à moyen terme.

Les approches fondées sur la négociation concernent souvent, mais pas exclusivement, des agents compétitifs et/ou rationnels (au sens de la rationalité économique). Chaque agent possède une stratégie de prise de décision qu'il utilise pour maximiser sa *fonction d'utilité locale*. Dans le cas où l'agent pourrait définir une stratégie dominante, c'est-à-dire une stratégie gagnante indépendamment des stratégies des autres, il n'y a pas réellement besoin de coordination. Cependant, cette stratégie n'existe jamais puisque les stratégies individuelles peuvent s'influencer mutuellement (si cette stratégie existait, elle serait applicable par tous les agents). L'agent, à travers sa rationalité individuelle, est amené à se coordonner avec les autres au moyen d'un protocole de négociation, afin de converger vers des solutions globales acceptables par tous les participants à la négociation.

Les protocoles proposés dans ce contexte cherchent à satisfaire généralement deux critères : 1) garantir l'efficacité globale sans affaiblir aucune fonction d'utilité locale ; 2) veiller à la stabilité du système, c'est-à-dire assurer un équilibre symétrique entre agents où les gains sont répartis équitablement. Les travaux les plus significatifs du domaine proposent des modèles de négociation inspirés de la théorie des jeux, de la théorie d'aide à la décision ou combinent les deux.

4.5.4.1. *Travaux basés sur la théorie des jeux*

Dans le domaine de la théorie des jeux, l'objectif est de trouver une solution sous la forme d'un *compromis* acceptable par tous les participants au jeu. Pour atteindre un tel compromis (appelé souvent *consensus*), chaque agent participant au jeu dispose d'un vecteur d'utilité (ou une fonction d'utilité qui agrège les différents paramètres du vecteur d'utilité) à partir duquel il lui est possible de déterminer un ensemble de situations du jeu correspondant à un accord potentiel avec les autres agents participants. La solution trouvée doit satisfaire plusieurs critères tels que le critère d'efficacité Pareto (*Pareto efficiency*). Une solution *S* est dite *Pareto*

optimale s'il n'existe pas de solution possible S' telle que S' soit meilleure que S pour tous les agents et qui soit strictement meilleure pour au moins l'un d'entre eux.

Les questions qui se posent alors sont : comment atteindre une solution qui privilégie tous les agents sans en affaiblir aucun ? et si une telle solution existe, est-elle unique ? Plusieurs travaux en théorie des jeux [KRE 90, NAS 50] ont proposé des méthodes pour trouver de telles solutions tout en spécifiant avec précision les *conditions de leur applicabilité*. A titre d'exemple, on peut citer l'équilibre de Nash dont l'intérêt principal est de garantir que, sous *les conditions de rationalité individuelle et de symétrie des informations*, les joueurs atteignent nécessairement un compromis qui maximise la fonction d'utilité de chaque participant [NAS 50].

Dans le cas d'agents compétitifs et en l'absence de stratégies dominantes, une solution acceptable par les agents est une solution qui n'affaiblit aucun agent. Dans [ZLO 94], Zlotkin et Rosenschein modélisent les dépendances entre agents *via* un modèle quantitatif basé sur les fonctions d'utilité. Ils présupposent l'existence d'un protocole de négociation (décidé par les concepteurs des agents sous la forme de règles) et, de là, chaque concepteur peut doter son agent d'une stratégie de prise de décision de son choix. Trois classes de domaines ont été explorées :

- domaine orienté tâche : il n'y a pas de conflit concernant les tâches et la négociation porte sur leur redistribution pour maximiser les bénéfices des agents ;
- domaine orienté état : la négociation porte sur les plans et leurs ordonnancements ;
- domaine orienté bénéfice (le mot d'origine est *worth*) : les agents négocient la redistribution de buts, sachant qu'un but est évalué par une fonction d'utilité.

Appliqués au domaine de la coordination dans les SMA, les modèles issus de la théorie des jeux fournissent des représentations intéressantes pour la négociation modélisée comme un jeu où les intérêts individuels des agents sont représentés par des fonctions d'utilité (une fonction d'utilité est éventuellement une agrégation de plusieurs critères). De plus, il est possible de garantir, sous les conditions théoriques du cadre de la théorie des jeux, la convergence des processus de négociation vers des consensus où aucune fonction d'utilité n'est affaiblie.

Cependant, relâcher ces conditions pour une meilleure adéquation aux SMA (par exemple : informations non symétriques, agents susceptibles de frauder et incertitude sur les informations reçues, etc.) est tout sauf trivial. En effet, dès que l'on quitte le cadre théorique bien délimité pour aborder des situations complexes, ni l'existence ni l'unicité d'une solution ne sont garanties, ce qui rend la généralisation de ces modèles quasiment impossible du fait de leur explosion combinatoire ou encore de leur incapacité à garantir la terminaison des processus de négociation ou la convergence vers des consensus satisfaisants (solutions quantitatives et qualitatives).

Ainsi, en faisant abstraction des travaux qui s'inspirent de ces modèles et qui se concentrent sur la recherche de solutions pour la stabilité des coalitions, la répartition des bénéfices entre agents, etc., la définition de modèles de négociation adaptés aux SMA reste une problématique intéressante pour la recherche dans ce domaine.

4.5.4.2. Travaux basés sur la théorie d'aide à la décision

Il existe également d'autres approches basées sur la théorie d'aide à la décision. Ces approches se fondent essentiellement sur des modèles de décision multicritères [ELF 00, MOR 96, MOR 99] pour la coordination d'agents.

A ce titre, on peut citer l'approche de négociation proposée par Pavlos Moraïtis et Alexis Tsoukiàs dans [MOR 99]. Cette approche consiste à coordonner des plans multi-agents. Basée sur un cycle de planification, négociation et exécution (PNE). La coordination des actions collectives est décidée durant les phases de planification et de négociation. Dans cette approche, trois types de buts sont distingués : les buts globaux partagés par les agents, les buts locaux des agents mais nécessaires au buts globaux et les buts *privés* propres aux agents qui n'ont aucun impact sur les buts globaux. Les agents coordonnent dynamiquement leurs plans pour achever un plan global à travers les buts locaux, mais en même temps essaient de satisfaire leurs buts privés exprimés sous la forme de *préférences individuelles*. Les agents développent leurs plans sur la base d'un modèle multicritère dont l'intérêt est double. D'une part, les agents peuvent évaluer leurs actions ou leurs plans en utilisant un ensemble de critères intégrant leurs buts privés. D'autre part, les agents peuvent mesurer l'utilité du plan pour atteindre leurs buts aussi bien privés que locaux (étape de planification). Ensuite, les plans élaborés sont échangés entre agents qui initient la phase de négociation. Cette phase est utile pour détecter les conflits potentiels, les résoudre et aussi pour créer des «*situations de coopération*» si les agents découvrent que leur collaboration améliore leurs plans mutuels.

Pour la résolution de conflit, la coordination est basée sur un mécanisme de négociation entre agents relativement à l'échelle des critères utilisés, l'ensemble des évaluations des critères, et les plans proposés ou même le modèle lui-même. La négociation est envisagée pour les agents deux à deux. Il y a un temps limité pour l'exécution jointe et les agents ont intérêt à coopérer puisque l'échec du but global est beaucoup plus néfaste pour tous les agents que s'ils concluaient un accord même s'il n'est pas optimal. Enfin, la phase d'exécution parallèle des plans commence dès qu'il n'y a plus d'objection – à propos des plans générés – de la part des agents impliqués ou si la phase de négociation est terminée.

EXEMPLE.— Dans le scénario des agents déménageurs, les préférences des agents peuvent porter sur deux critères :

- (max-profit, p) : l'agent cherchera à maximiser le profit lié au transport des objets ;

– (min-temps, t) : l'agent cherchera à minimiser le temps consommé pour le transport des objets.

Supposons par exemple que les actions :

– Id.Déplacer(X), Id.Soulever(X) apportent à l'agent un profit de deux unités chacune, alors qu'elles lui font perdre 1 unité de temps chacune ;

– Id.Transporter(X), Id.Charger(X) apportent à l'agent un profit de trois unités chacune, alors qu'elles lui font perdre deux unités de temps chacune ;

– Id.Attendre() fait perdre une unité de temps et une unité de bénéfice.

Chaque agent génère son plan sous la forme d'un graphe d'actions. Il va ensuite générer tous les chemins efficaces possibles (efficient paths) de son plan. Pour cela, il applique son modèle de préférences multicritères. Les agents échangent ensuite leurs meilleurs plans qui représentent le chemin le plus efficace pour que chaque agent atteigne son but en satisfaisant au mieux ses préférences. A partir des plans échangés, chaque agent calcule le graphe représentant le plan collectif efficace (optimal pour tous) de négociation, appelé le graphe de négociation et qui respecte le modèle multicritère.

L'intérêt majeur de cette approche est qu'elle introduit une représentation basée sur un graphe multicritère des plans des différents agents, ce qui en fait une méthode distribuée sur les agents. En outre, elle repose sur un modèle multicritère pour la phase de négociation.

4.5.5. Modèles de coordination fondés sur la formation de coalitions

Les coalitions permettent aux agents de satisfaire des besoins qui requièrent la synergie de compétences des différents agents comme, par exemple, dans le cadre de la résolution de tâches complexes pour lesquelles des agents agissant seuls seraient incapables ou moins efficaces (par exemple : déménagement d'un objet lourd). Une coalition peut être définie comme une organisation à court terme basée sur des engagements spécifiques et contextuels, ce qui permet aux agents de bénéficier de leurs compétences respectives. Dans un contexte économique par exemple, plusieurs compagnies s'allient « *virtuellement* » pour répondre à des offres nécessitant des compétences variées. La formation de coalitions offre plusieurs avantages : 1) le concept d'engagement ponctuel permet aux agents de réagir de façon opportuniste et de réviser dynamiquement leurs intérêts et par conséquent leurs objectifs ; 2) la formation et la dissolution des coalitions est contexte-dépendante, ce qui permet aux agents d'adapter leur production ; 3) contrairement aux organisations statiques, la formation de coalitions permet d'appréhender de façon plus flexible les environnements ouverts et dynamiques.

Du point de vue des SMA, les recherches sur la formation de coalitions essaient de proposer des protocoles d'automatisation des coalitions qui soient les plus « réalistes » possibles. En effet, la recherche de solutions optimales (généralement mesurées en termes de profit total maximum) nécessite des algorithmes NP-complets ; et à cause de cette complexité, différentes hypothèses simplificatrices sont introduites pour générer des solutions calculables en un temps raisonnable. Une des hypothèses les plus fortes consiste à supposer que les agents sont coopératifs, voire altruistes. Or dans un contexte de rationalité économique, les agents sont libres de se comporter comme ils le désirent, et même d'avoir des objectifs égoïstes comme en commerce électronique.

Cependant, la majorité des travaux en SMA qui se sont intéressés aux organisations dynamiques et en particulier à la formation de coalitions supposent *a priori* que tous les agents ont la même rationalité voire la même stratégie, ce qui permet de réduire considérablement l'espace de recherche. Dans [SHE 95] par exemple, le problème est simplifié en un calcul distribué de la meilleure solution pour le système comme un *tout*. Notons que même sous cette hypothèse, le problème reste difficile car un agent n'est pas sensé avoir une vision globale du système.

Le problème de la formation de coalitions consiste à trouver une solution qui soit la plus satisfaisante pour l'ensemble des agents. C'est typiquement le cas lorsque l'on cherche à calculer la valeur de Shapley qui correspond *grosso modo* à l'espérance de l'utilité : on cherche la configuration la plus satisfaisante en moyenne. Certes, l'intérêt global du SMA est pris en compte, mais la méthode imposée laisse peu de place aux préférences individuelles des agents et par conséquent à leur autonomie. En effet, soit la configuration est calculée de façon externe, soit elle est calculée par les agents eux-mêmes, mais en leur attribuant un rôle bien défini qu'ils doivent jouer. C'est le cas par exemple dans [ZLO 94] qui propose une méthode de calcul distribuée de la valeur de Shapley. Les auteurs imposent donc aux agents que la meilleure solution soit celle qui maximise la valeur de Shapley. Certes, pour garantir une relative satisfaction de l'agent, ils prennent soin de ne pas le forcer à former une coalition qui ne lui apporte pas plus que ce qu'il aurait eu sans participer à la coalition. Mais ce déterminisme ne permet pas aux agents de développer une stratégie propre (par exemple : si un agent accepte de prendre le risque de créer une coalition à faible rendement dans l'espoir de former plus tard une coalition très productive, cela peut lui être très bénéfique).

Dans [KET 94], Steven Ketchpel a redonné un peu de liberté aux agents en leur permettant d'avoir des estimations différentes du revenu d'une coalition. Le problème essentiel est alors la répartition des bénéfices qui ne sont pas connus avec précision au moment de la formation de la coalition. L'algorithme proposé consiste en une suite d'agrégations d'agents et de coalitions. Chacun négocie son prix de participation en fonction de ses croyances sur la valeur de la coalition. Dans un

deuxième temps, les agents décident comment diviser les revenus futurs de la coalition. Pour chaque coalition, les agents élisent un *manager* qui négocie pour eux. Cette élection consiste en fait en une vente aux enchères où celui qui promet le plus est élu. Ce qui est intéressant ici, c'est que les agents peuvent avoir des conceptions différentes des revenus et parvenir tout de même à un accord. Mais, comme beaucoup d'autres, cet algorithme suppose que tout est monnayable, que tout peut être estimé par une valeur réelle, ce qui a des conséquences sur l'algorithme lui-même, le rendant ainsi très dépendant de ce présupposé. C'est pour éviter ceci que d'autres travaux ont fait le choix de préférences indépendantes des critères qui les ont produites [VAU 00], laissant aux agents toute la liberté pour élaborer leurs propres stratégies.

Une autre critique qui peut être adressée aux travaux existants sur la formation de coalitions concerne les dépendances entre les coalitions telles que définies dans [KET 94] : dans une formation itérative, un agent ne peut pas prendre en compte la participation future éventuelle de tel ou tel agent. Or, l'intérêt pour une coalition dépend de l'ensemble de ses membres et non seulement de l'augmentation de sa valeur à l'étape en cours. De même, très peu de travaux se sont intéressés aux partitions multiples dans le processus de formation de coalition. Un agent est le plus souvent contraint de n'appartenir qu'à une seule coalition à la fois (partitions primaires). Ce qui simplifie considérablement le problème mais très souvent le dénature [SEN 00].

Des travaux sur la formation de coalitions plus récents ont porté sur la recherche de consensus par formation de coalitions. On peut citer l'approche proposée dans [AKN 00], qui consiste à coordonner des agents lors de l'allocation de tâches. Les agents considérés peuvent être coopératifs ou compétitifs. Cette approche se fonde sur une agrégation multicritère basée sur des techniques d'aide à la décision. Une autre approche très originale a été proposée par [VAU 00]. Elle tente de s'affranchir des limites classiques des protocoles de consensus existants, telles que l'application d'une stratégie de décision connue de tous les agents, le partage des modèles de préférences ou des critères à optimiser, etc. Dans cette approche, les auteurs proposent un algorithme de consensus qui privilégie *les libertés individuelles* des agents [VAU 01]. En effet, chaque agent dispose de sa propre représentation de la rationalité économique (c'est-à-dire la rationalité individuelle) et ses critères ne sont pas connus des autres. De plus, un agent peut changer de stratégie durant la phase de recherche de consensus : il peut changer ses critères et leur pondération, ses préférences, etc. Le protocole proposé est distribué, garantit l'équité entre agents (c'est-à-dire la symétrie des informations) et favorise leur liberté : seules les préférences des agents sont échangées et aucune indication sur le calcul symbolique ou numérique des préférences n'est perceptible des autres agents. Enfin, l'algorithme proposé garantit la convergence vers un consensus grâce à l'introduction du concept d'alliances. Une alliance est la coalition minimale qui

regroupe deux agents ayant des préférences jugées les plus proches (par rapport à une distance entre les préférences des agents).

4.6. Coordination d'actions par planification multi-agent et distribuée

Dès le début des années quatre-vingt, plusieurs travaux en coordination ont adopté la planification multi-agent comme technique de coordination des activités des agents [DUR 88, GEO 83]. La plupart de ces travaux se sont focalisés sur la planification [GEO 90] comme stratégie de coopération entre agents cognitifs pouvant assumer une *postplanification*. En effet, souvent, les agents génèrent d'abord leurs plans et cherchent à les coordonner ultérieurement en vue de les exécuter. L'obtention d'un plan global cohérent nécessite la détection de sous-buts en interactions positives ou négatives et, ensuite, la résolution des conflits potentiels. Dans la suite, nous donnons un panorama des principales approches de coordination par planification multi-agent proposées dans la littérature SMA.

4.6.1. La coordination par synchronisation d'actions

Cette approche s'apparente aux techniques utilisées dans les systèmes distribués. Il s'agit de spécifier l'enchaînement des actions et de gérer leur parallélisme potentiel. Cette technique n'intègre pas de mécanismes cognitifs et peut être vue comme un support de base à toute autre forme de coordination d'actions. A titre d'exemple, on peut citer les travaux de M. Georgeff [GEO 83]. Celui-ci a introduit un agent *synchroniseur* qui gère les interactions entre plans. Les agents envoient leurs plans au *synchroniseur*. Ce dernier les examine, détecte les régions critiques dans les plans reçus et résout les conflits par l'ajout d'actions de synchronisation (par exemple, des primitives de synchronisation qui garantissent l'exclusion mutuelle entre les actions conflictuelles).

4.6.2. La planification multi-agent comme spécialisation de la résolution distribuée de problème

Dans [DUR 99], la planification multi-agent est présentée comme une *spécialisation* de la résolution distribuée de problème. En effet, dans cette perspective, chaque agent se charge de la résolution d'un sous-problème qui, dans le cas d'une planification multi-agent, devient un problème de planification : génération d'un sous-plan pour un sous-problème. De nouveaux problèmes sont alors posés aux agents planificateurs (voir ci-dessus : scénario des déménageurs) :

- Comment décomposer un problème en sous-problèmes ? Quelle stratégie le gestionnaire de l'agence doit-il adopter pour décomposer efficacement la tâche T en sous-tâches ?
- A quels agents allouer ces sous-problèmes ? Quelle stratégie le gestionnaire de l'agence doit-il adopter pour allouer efficacement les sous-tâches aux agents déménageurs ?
- Comment générer les sous-plans ? Chaque agent déménageur génère un plan en fonction de ses connaissances (description du monde incluant les connaissances sur les autres agents, l'état courant, l'état final à atteindre, etc.) ;
- Quand échanger les solutions ou les résultats partiels ? Chaque fois qu'un agent détecte (ou prédit) une situation d'interaction positive ou négative, il en informe les agents concernés et entre en phase de coordination ;
- Quand et comment faire la synthèse des plans concurrents ? Pour des environnements dynamiques, il est préférable voire indispensable d'entrelacer synthèse de plans et exécution. En effet, un plan à long terme peut s'avérer obsolète car, contrairement à la planification mono-agent, l'hypothèse du monde clos n'est pas viable dans un contexte SMA.

4.6.3. La coordination par planification multi-agent centralisée ou distribuée

Les agents élaborent un *plan multi-agent* qui spécifie leurs actions futures et leurs interactions. La synthèse des plans exige un raisonnement sur la manière dont les actions des différents agents peuvent interférer et, par conséquent, la représentation d'actions parallèles. L'exécution non contrôlée de plans concurrents peut engendrer des situations de blocage dues par exemple à une utilisation incohérente des ressources (par exemple : le couloir étroit est une ressource critique dans le scénario des agents déménageurs). De ce fait, la synchronisation des plans doit être effectuée à *différents points* de la résolution d'un problème : durant la phase de décomposition du problème [COR 79], lors de la construction des plans en bâtissant des plans hiérarchisés qui interagissent faiblement [COR 79], après la construction des plans [GEO 83, KAB 95], en alignant des plans partiels de façon incrémentale [DEC 92, DUR 91], ou en introduisant dans le processus de planification un mécanisme de raisonnement sur les interactions et les dépendances possibles entre actions et plans [ELF 96b, EPH 95b, ROS 82, VMA 92].

4.6.4. Planification multi-agent centralisée et plans distribués

Ici, le *plan multi-agent* est généré de manière centralisée par un agent *coordinateur* et son exécution est distribuée sur les agents. De ce fait, le plan généré

est nécessairement un plan partiellement ordonné et les actions parallèles sont exécutées par différents agents de manière concurrente.

Après avoir distribué le plan en sous-plans potentiellement synchronisés, le coordinateur les transmet aux agents exécutants en vue de leur achèvement sous la forme de plusieurs processus concurrents. Dans le scénario des agents déménageurs, cette situation correspondrait au cas où l'agent gestionnaire allouerait à chaque agent un plan correspondant à la sous-tâche qu'il doit effectuer.

Les systèmes proposés en *planification multi-agent centralisée* [GEO 83, GEO 88] présentent des options intéressantes pour la résolution des conflits et la convergence vers une solution globale, mais présentent tous les mêmes désavantages liés à la centralisation du contrôle au niveau d'un seul agent, à savoir :

- si le coordinateur est défaillant, le système est incontrôlable ;
- le coordinateur central constitue un goulet d'étranglement au niveau de la communication et du traitement, c'est-à-dire que les différents agents travaillant de façon concurrente dans le réseau doivent envoyer leurs actions au coordinateur afin qu'il résolve les conflits éventuels et gère leurs interactions ;
- la charge de communication est très importante puisque des quantités importantes de données non nécessairement utiles sont envoyées vers le coordinateur en pénalisant le débit utile du réseau.

4.6.5. *Planification multi-agent distribuée*

En *planification multi-agent distribuée*, c'est le processus de la synthèse du plan multi-agent qui est distribué. A nouveau, on peut distinguer deux cas de figure selon que le plan multi-agent est centralisé ou non.

Planification distribuée pour plan centralisé

Dans certains contextes (par exemple celui de la planification pour des problèmes complexes) la synthèse de plans requiert plus d'un agent. Dans ce cas, le plan sera élaboré par plusieurs agents planificateurs. Reprenons l'exemple des agents déménageurs qui doivent planifier la tâche T. Après la décomposition de la tâche en sous-tâches, chaque agent responsable d'une sous-tâche génère le sous-plan correspondant et, ensuite, les agents interagissent pour fusionner les sous-plans en un plan global. L'interaction peut concerner l'échange de sous-plans à synchroniser, des sous-plans partiels à raffiner, inclure un mécanisme de négociation, etc. De nouveau, le coût de communication peut être important et sans garantie de convergence vers un plan global. Toutes les techniques utilisées en résolution distribuée de problèmes restent applicables pour résoudre les conflits éventuels entre agents planificateurs.

Planification distribuée pour plan distribué

Contrairement aux approches précédentes, ici les processus de synthèse et d'exécution d'un plan multi-agent sont distribués.

On distingue à nouveau deux approches : une approche *orientée tâches* et une approche *orientée agents*. La différence concerne l'existence ou non d'un but global.

Dans le cas d'une approche *orientée tâche*, le problème est décomposé en sous-problèmes et chaque agent prend en charge la résolution du sous-problème qui lui a été confié, c'est-à-dire qu'il génère un sous-plan et l'exécute. La synthèse des sous-plans porte sur les *résultats* et non plus sur les plans.

L'approche *orientée agents* (souvent appelée planification distribuée) s'intéresse aux systèmes où plusieurs agents peuvent planifier et exécuter des plans de manière concurrente et autonome [EPH 95a, GEO 90, VMA 92] indépendamment de l'existence d'un plan global ou d'un but global au système. La coordination, ici, consiste à fournir les mécanismes nécessaires pour que l'exécution des plans concurrents soit possible.

Comme les agents planifient dans un environnement partagé, des événements liés à l'exécution concurrente de plans peuvent se produire durant les phases de synthèse ou d'exécution de plans. Par conséquent, la planification multi-agent requiert des modèles d'actions et de plans multi-agents [GEO 90] permettant la gestion de la concurrence potentielle entre les activités des agents. Ceci a conduit à l'élaboration de formalismes offrant une grande puissance d'expression du parallélisme, de concurrence d'actions, d'événements non atomiques, etc., ainsi que l'expression de nouveaux buts tels que les buts de maintenance ou d'ordonnancement d'actions.

Différentes approches ont été proposées en planification distribuée et plus particulièrement en planification *incrémentale*. Celle-ci consiste à considérer un ensemble de plans déjà coordonnés dans lequel on insère un nouveau plan à coordonner avec les plans existants.

L'approche développée par F. Von Martial [VMA 92] repose sur la définition d'une taxonomie des relations entre les plans de différents agents et le développement d'une structure de communication entre des agents autonomes qui échangent leurs plans et négocient pour résoudre les relations entre ces plans. Le modèle formel proposé offre plusieurs avantages, tels que la résolution des interactions positives et négatives. Cependant, il ne coordonne que deux agents à la fois, ce qui ne permet pas de traiter les interactions entre n agents.

L'approche développée par Alami [ALA 94] est fondée sur le paradigme de fusion de plans. Le modèle proposé est robuste et coordonne n agents à la fois, mais la coordination est contrôlée par le nouvel agent qui insère son plan. Le nouveau plan est généré à partir des plans existants, ce qui est incompatible avec la notion d'agents autonomes qui produisent leur plan indépendamment de ceux des autres et les coordonnent ensuite.

L'approche présentée par A. El Fallah et S. Haddad dans [ELF 96a] propose un algorithme de planification distribué qui coordonne n agents simultanément. Le modèle formel est basé sur la notion d'ordre partiel entre les actions. La coordination de plans consiste à introduire des arcs de synchronisation (sorte de liens causaux) entre les actions en interaction. D'une part, l'algorithme proposé résout les interactions positives en utilisant les pré- et postconditions des actions. D'autre part, il résout les interactions négatives par synchronisation d'actions. Le modèle garantit la faisabilité du plan multi-agent, et ce quel que soit l'ordre total qui étend l'ordre partiel. Cependant, le formalisme utilisé implique une représentation du monde très abstraite et le traitement des interactions positives ne couvre qu'un nombre restreint de situations.

Une extension de ce modèle a consisté à proposer un formalisme de représentation de plan plus riche [ELF 96b]. Il s'agit d'une extension stricte des réseaux de Petri [JEN 91] appelée *réseaux de Petri récursifs* (RdPR) [ELF 95, ELF 96b]. Le modèle de planification proposé unifie l'algorithme de coordination à n agents [ELF 96a] et le formalisme des RdPR. Ce dernier, tout en étant indépendant du domaine, recouvre les principales caractéristiques de la planification distribuée (la récursivité, la dynamique, l'entrelacement entre planification et exécution, etc.). Il permet de représenter et de raisonner sur des actions simultanées et des processus continus (actions concurrentes, choix parmi plusieurs alternatives, synchronisation, etc.).

Ce modèle de planification permet la représentation d'actions abstraites au moment de la synthèse de plans. Le raffinement d'une action abstraite est dynamique (en cours d'exécution du plan) et tient compte du contexte courant d'exécution (plusieurs alternatives sont associées à une même action abstraite), ce qui permet de choisir dynamiquement la façon dont on exécute une action. De plus, il permet de vérifier la consistance des plans produits à l'aide d'algorithmes efficaces. La coordination de plans est réalisée à travers des opérations de fusion, de séquentialisation et de parallélisation de plans, et supporte la résolution des interactions positives et négatives avec la garantie de la consistance des traitements.

4.6.6. *Processus de coordination d'actions par planification*

Un processus de coordination par planification peut être décrit par le cycle suivant :

- 1) caractérisation des situations d'interaction,
- 2) détection *statique ou dynamique* des situations d'interaction,
- 3) résolution des situations d'interaction positives et négatives,
- 4) modification des plans.

4.6.6.1. *Caractérisation des situations d'interaction*

Dans le cas de la coordination par planification, les situations d'interaction sont de deux types :

- situations d'interaction positives : on cherchera à favoriser la synergie entre les actions et plans des agents ;
- situations d'interaction négatives : on cherchera à éviter les conflits entre actions et plans. Ces conflits se ramènent le plus souvent à un problème de gestion de ressources.

On peut trouver des caractérisations assez fines de ces types d'interaction dans la littérature [DEC 92, ELF 96b, VMA 92]. Nous reprenons ici quelques situations présentées dans [ELF 96b]. Pour simplifier la présentation, nous allons réutiliser la modélisation des actions proposée dans le paragraphe 4.4.2, c'est-à-dire que chaque action a des préconditions dont la satisfaction est nécessaire à son exécution. L'exécution d'une action rend valide ses postconditions et que les pré- et postconditions sont exprimées sous la forme de prédicats avec des littéraux contenant des variables représentant des objets du domaine.

4.6.6.1.1. Interactions positives

Les interactions positives traduisent des situations où les actions d'un agent facilitent l'exécution du plan d'un autre agent. Nous distinguons deux types d'interactions positives.

Deux actions sont dites *redondantes* si elles apparaissent dans deux plans appartenant à deux agents différents, ont les mêmes pré- et postconditions, et si leurs instanciations sont équivalentes. Dans ce cas, le processus de la coordination affecte l'exécution de l'action à l'un des deux agents. Moyennant une synchronisation entre les deux plans *via* l'action commune, l'agent chargé de l'action l'exécute et en fournit les résultats à l'autre qui les intègre dans son plan. Dans l'exemple des déménageurs (voir figure 4.5) : les actions A_1 .Déplacer (Lampe) et A_2 .Déplacer (Lampe) sont redondantes.

Deux actions sont dites *favorables* si l'exécution de l'une valide totalement ou partiellement les préconditions de l'autre. L'exécution de cette dernière sera respectivement possible ou favorisée. Dans ce cas, le processus de coordination synchronise (par exemple : via un lien causal) les deux actions de sorte que l'action favorable soit exécutée avant celle favorisée et qu'il n'y ait pas d'autres actions entre les deux qui invalident les postconditions de l'action favorable. Dans l'exemple des agents déménageurs (voir figure 4.5) : l'action A_1 .Soulever(Table) est favorable à l'action A_2 .Déplacer(Table). L'inverse n'est pas forcément vrai.

4.6.6.1.2. Interactions négatives

Les interactions *néglatives* traduisent des situations où les actions d'un agent empêchent momentanément ou définitivement l'exécution du plan d'un autre agent. A titre d'exemple, nous distinguons ici trois types d'interactions négatives.

EXEMPLE.— Si on raffine l'action A.Transporter(X) (voir figure 4.6) selon le squelette donné dans la figure (voir figure 4.7, partie de gauche), on obtient après instantiation les deux plans pour A_1 et A_2 (voir figure 4.7, partie de droite) pour les deux actions : A_1 .Transporter(Table) et A_2 .Transporter(Tapis).

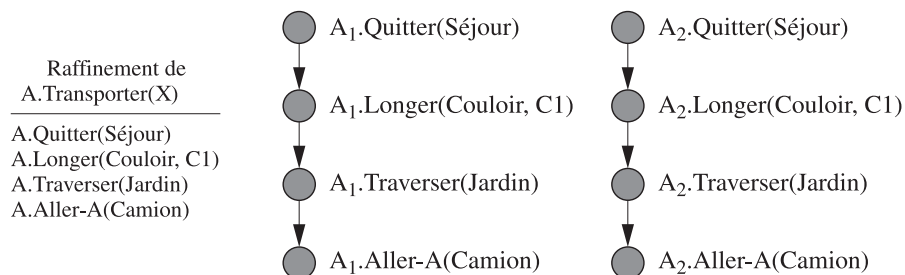


Figure 4.7. Plans raffinés de A_1 et A_2

Supposons que la représentation du monde contiennent la proposition suivante : Capacité(Couloir, C1) = 1). Dans ce cas, que les actions A_1 . Longer(Couloir, C1) et A_2 . Longer(Couloir, C1) peuvent être en conflit (mutuellement défavorables) si elles sont simultanées. Pour résoudre ce type de conflit, le mécanisme de coordination doit empêcher l'exécution simultanée des deux actions. Dans le modèle proposé par F. Von Martial par exemple, le formalisme de représentation des plans gère les intervalles temporels associant des durées aux actions.

Il propose plusieurs heuristiques pour gérer les conflits entre actions : déplacement d'intervalles d'actions, réduction d'intervalles d'actions, etc. (voir figure 4.8).

Deux actions sont dites *défavorables* si l'exécution de l'une invalide totalement ou partiellement les préconditions de l'autre. Ainsi, l'exécution de cette dernière sera respectivement impossible ou désavantagee. Une telle situation doit être détectée *a priori* afin de prédire d'éventuels échecs ou blocages. A nouveau, le mécanisme de coordination introduit un ordre entre les actions défavorables pour garantir leur exécution.

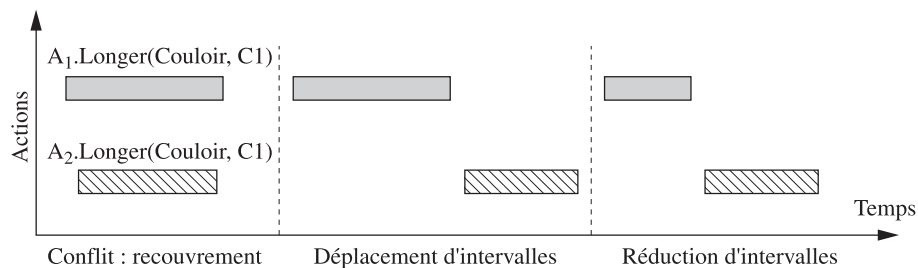


Figure 4.8. Heuristiques pour la résolution de conflits entre A_1 et A_2

Deux actions sont dites *exclusives* si l'exécution de l'une empêche momentanément l'exécution de l'autre : leur exécution requiert une même ressource non consommable (par exemple : couloir1 de la figure 4.8). Ici aussi, le mécanisme de coordination introduit un ordre entre les actions défavorables. D'autres heuristiques peuvent être considérées comme dans [VMA 92] pour gérer ce type de situations (substitution de ressources, déplacement d'intervalles d'utilisation de ressources, etc.).

Deux actions sont dites *incompatibles* si l'exécution de l'une empêche définitivement l'exécution de l'autre (par exemple si leur exécution requiert une même ressource consommable). La coordination consiste à rendre les deux exécutions possibles, par exemple en opérant une substitution des ressources (forcément critiques dans ce cas), ou à trouver une action alternative à l'une ou l'autre des actions en conflit.

A partir des situations d'interaction entre actions, on peut généraliser aux interactions entre plans. Par exemple, deux plans sont favorables s'ils contiennent au moins deux actions favorables et aucune interaction négative, etc. Pour plus de

détails sur les heuristiques de coordination en planification distribuée, le lecteur est invité à consulter [DEC 92, DUR 91, ELF 96a, VMA 92].

4.6.7. Exécution concurrente de plans et dynamique de l'environnement

La prise en charge de la dynamique de l'environnement requiert une détection à la fois statique et dynamique des situations de coordination si on veut éviter que les agents passent leur temps à replanifier leurs activités.

En effet, la planification à long terme n'est pas viable car les plans générés à l'avance risquent de devenir plus ou moins obsolètes dans un environnement qui change fortement, et aussi du fait de la vision partielle des agents. Cependant, dans la plupart des approches, les situations d'interaction sont détectées statiquement [DEC 92, DUR 91, VMA 92]. En général, les approches de coordination les plus utilisées sont les suivantes.

Postcoordination de plans

Chaque agent formule un ensemble de plans possibles (plusieurs alternatives) et la coordination consiste à trouver la bonne combinaison qui garantit l'exécution du plan global. Cette méthode n'est pas viable pour deux raisons : la complexité du choix et le risque d'explosion combinatoire du nombre de combinaisons. De plus, cette méthode suppose que les agents connaissent tous les plans détaillés les uns des autres.

Précoordination de plans

Avant de planifier, certaines contraintes de planification sont imposées/échangées aux/entre agents, de sorte que les plans produits soient compatibles. Dans ce cadre, on retrouve les lois sociales à respecter, des règles de coopération à adopter, etc. Cela suppose toutefois que les agents planifient dans un contexte respectant une certaine organisation et qu'ils respectent les règles de comportements collectifs. C'est ainsi que, sous cette hypothèse, certains travaux considèrent que l'organisation dans les SMA peut être considérée comme un moyen de coordination à long terme. Même en admettant cette hypothèse, ces approches sont incapables de prendre en compte l'évolution spontanée de l'environnement.

Entrelacement de la planification et l'exécution

L'entrelacement entre planification et exécution de plans peut être d'un grand secours dans le cas des environnements instables [ELF 96b, ROS 96]. Mais dans ce cas, la détection des situations d'interaction doit être statique et dynamique à la fois. Dans [ELF 96b] par exemple, une situation d'interaction positive peut se présenter lors de la phase de raffinement (et donc en cours d'exécution) d'une action abstraite

(par exemple si une des actions du plan requiert une compétence non locale). Les interactions négatives sont détectées lorsque des actions appartenant à des plans différents manipulent des objets partagés de manière concurrente (voir figure 4.8).

4.7. Critères d'évaluation des systèmes de coordination

Les techniques et mécanismes de coordination peuvent être classifiés selon le spectre de l'autonomie, de la flexibilité, du degré de contrôle de la coordination [LES 91]. L'évaluation d'un modèle de coordination se base sur les critères suivants ou une de leurs combinaisons :

- efficacité en termes de solutions atteintes ou non (complétude) ;
- qualité des solutions produites (consistance) et leur opportunité ;
- adaptabilité du système (apprentissage de concepts individuels ou de représentations et modèles des autres, etc.) et gestion de l'incertain ;
- clarté et observabilité du système ;
- fonctionnement en mode dégradé, tolérance aux pannes, etc. ;
- réactivité aux changements, degré et coût des informations échangées, etc. ;
- etc.

Modèles Critères	Résolution distribuée de problèmes	Organisations	Protocoles de coopération	Planification
1. Efficacité	Oui	Oui	Oui	Oui
2. Qualité de solution	Oui	Oui	Oui	Oui
3. Adaptabilité	Non	Oui si organisation dynamique	Non	Oui si entrelacement
4. Clarté	Non	Non	Oui	Oui
5. Tolérance	Non	Oui si organisation dynamique	Non	Oui si entrelacement
6. Réactivité	Non	Oui si organisation dynamique	Non	Oui si planification réactive

Tableau 4.1. Principales caractéristiques d'un modèle de coordination

Dans DVMT, par exemple, Lesser et son équipe ont conceptualisé le degré de coopération en mesurant la performance comme le nombre total des sources de connaissances activées par tous les nœuds de la collection quand le système atteint sa solution optimale. Plus la collection s'approche de la solution optimale (nombre d'activations connues) moins on gaspille d'effort et plus la performance est meilleure.

4.8. Conclusion

Les travaux développés dans le domaine de la coordination d'agents combinent souvent plusieurs des mécanismes présentés dans ce chapitre. Citons à titre d'exemple le *réseau de conscience* (awareness network) dans lequel il y a communication de toutes les actions accompagnées d'une diffusion de tous les changements d'états ; les travaux liant connaissances communes et communication [FAG 95], ou encore les travaux en planification multi-agent qui incluent des mécanismes de prise de décision distribuée et de négociation.

Globalement, la plupart des approches de coordination se basent sur la distribution du contrôle et des données. Cette distribution est inhérente à l'architecture intrinsèque d'un SMA. Le contrôle distribué signifie que chaque agent possède un minimum d'autonomie pour générer de nouvelles actions et décider quels buts il va poursuivre. En revanche, cette distribution induit une dispersion des connaissances et une vue partielle rendant ainsi la cohérence globale difficile à atteindre ; d'où la nécessité de la coordination qui reste un mécanisme-clé dans le domaine des SMA. Plusieurs solutions ont été explorées et plusieurs ouvertures restent à l'état d'investigation. En particulier, peu de systèmes intègrent le concept d'apprentissage, dans leurs mécanismes de coordination, pour renforcer l'adaptabilité des agents. Beaucoup d'efforts sont également à fournir pour développer des mécanismes de coordination qui soient dynamiques et réellement distribués aussi bien pour des agents coopératifs que compétitifs.

4.9. Bibliographie

- [AKN 00] AKNINE S., PINSON S., SHAKUN M.F., « New coalition formation methods for multi-agent coordination », Jennings N.R., Sykara K., Wooldridge M. (dir.), *International Journal of Autonomous Agents and Multi-Agent Systems*, Kluwer, 2000.
- [ALA 94] ALAMI R., ROBERT F., INGRAND F., SUZUKU, S., « A Paradigm for Plan-Merging and its use for Multi-Robot Cooperation », dans *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Texas, 1994.
- [BON 88] BOND A.H., GASSER L. (dir), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers, 1988.

- [BRA 87] BRATMAN M.E., *Intentions, Plans and Practical Reason*, Harvard University Press, Cambridge, MA, 1987.
- [BRI 95] BRIGGS W., COOK D.J., « Flexible social laws », dans *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.
- [CAS 98] CASTELFRANCHI C., « Modelling Social Action for AI Agents », *Artificial Intelligence*, 103, p. 1-2, 157-182, 1998.
- [CAS 99] CASTELFRANCHI C., FALCONE R., « Basic Mental Attitudes of a Collaborating Agent: Cognitive Primitives for MAS », *MAAMAW 1999*, p. 188-209, 1999.
- [COR 79] CORKILL D.D., « Hierarchical Planning in Distributed Environment », dans *Proceedings of IJCAI'79*, p. 168-175, 1979.
- [DAV 83] DAVIS R., SMITH R., « Negotiation as a metaphor for distributed problem solving », *Artificial Intelligence*, n° 20, p. 63-109, 1983.
- [DEC 92] DECKER K.S., LESSER V.R., « Generalizing the Partial Global Planning Algorithm », dans *International journal on Intelligent Cooperative Information Systems*, p. 319-346, 1992.
- [DEC 93] DECKER K., LESSER V., « A one-shot dynamic coordination algorithm for distributed sensor networks », dans *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, p. 210-216, 1993.
- [DEC 95] DECKER K., LESSER V., « Designing a family of coordination mechanisms », dans *Proceedings of the First International Conf. on Multi-agents Systems (ICMAS-95)*, p. 73-80, 1995.
- [DIG 99] DIGNUM F., « Autonomous Agents with Norms », *Artificial Intelligence and Law*, vol. 7, n° 1, p. 69-79, 1999.
- [DUR 87] DURFEE E.D., LESSER V.R., CORKILL D., « Coherent cooperation among communicating Problem solvers », *IEEE transactions on computer*, C-36, vol. 11, n° 1275-1291, 1987.
- [DUR 87] DURFEE E.D., LESSER V.R., CORKILL D., « Cooperation Through Communication in a Distributed Problem Solving Network », dans M. Huhns (dir.), *Distributed Artificial Intelligence*, Pitman, 1987.
- [DUR 88] DURFEE E.D., *Coordination of Distributed Problem Solvers*, Kluwer Academic Press, Boston, 1988.
- [DUR 91] DURFEE E.D., MONTGOMERY A., « Coordination as Distributed Search in a Hierarchical Behavior Space », *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, SMC-21(6), p.1363-1378, 1991.
- [DUR 97] DURFEE E.D., KENNY P.K., KLUGE K.C., « Integrated Prepermission Planning and Execution for Unmanned Ground Vehicles », dans *Proceedings of the First International Conference on Autonomous Agents*, p. 348-354, 1997.
- [DUR 99] DURFEE E.D., « Distributed Problem Solving », dans G. Weiss (dir.), *Multi-Agent Systems: A modern approach to distributed artificial intelligence*, MIT Press, 1999.

- [ELF 95] EL FALLAH-SEGHROUCHNI A., HADDAD S., « A Formal Model for Coordinating Plans in Multi-agents Systems », dans *proceedings of Intelligent Agents Workshop*, Augusta Technology Ltd, Brooks University, Oxford, Grande-Bretagne, 1995.
- [ELF 96a] EL FALLAH-SEGHROUCHNI A., HADDAD S., « A Coordination Algorithm for Multi-agents Planning », dans *proceedings of MAAMAW'96*, Springer-Verlag, LNAI 1038, Eindhoven, Netherlands, 1996.
- [ELF 96b] EL FALLAH-SEGHROUCHNI A., HADDAD S., « A Recursive Model for Distributed Planning », dans *Proceedings of ICMAS'96 (International Conference of Multi-agents Systems)*, AAAI Press, Kyoto, Japon, 1996.
- [ELF 99] EL FALLAH SEGHROUCHNI A., MORAÏTIS P., TSOUKIÁS A., « Multi-Criteria Negotiation for Agent-Based Decision makers », *DSI'99. 5th International Conference on Decision Science Institute*, Athènes, 1999.
- [ELF 00] EL FALLAH SEGHROUCHNI A., MORAITIS P., TSOUKIAS A., « An aggeragation-Disaggregation Approach for Automated negotiation in Multi-Agent Systems », dans *Proceeding of MAMA'2000 (Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce)*, Wollongong, Australie, 2000.
- [EPH 95a] EPHRATI E., ROSENSCHEIN J.S., « A Framework for the Interleaving of Execution and Planning for Dynamic Tasks by Multiple Agents », dans *Proceedings of ATAL'95*, 1995.
- [EPH 95b] EPHRATI E., POLLACK E.M., ROSENSCHEIN S., « A tractable heuristic that maximizes global utility through local plan combination », dans *Proceedings of the First International Conf. on Multi-agents Systems (ICMAS-95)*, p. 94-101, 1995.
- [FAG 95] FAGIN R., HALPERN J.Y., MOSES Y., VARDI M.Y., *Reasonning about knowledge*, MIT Press, 1995.
- [FER 95] FERBER J., *Les Systèmes Multi-Agents: vers une intelligence collective*, Inter-Editions, 1995.
- [GAS 84] GASSER L., *The Social Dynamics of Routine Computer Use in Complex Organizations*, Ph.D. thesis, University of California, Department of Computer Science, Irvine, 1984.
- [GEL 92] GELERNTER D., CARRIERO N., « Coordination Languages », *Communications of the ACM*, 35, 2, 1992.
- [GEO 83] GEORGEFF M.P., « Communication and Interaction in Multi-agents Planning », dans *Proceedings of AAAI'93*, p. 125-129, 1983.
- [GEO 88] GEORGEFF M.P., « A Theory of Action for Multi-agents Planning », dans A.H. Bond et L. Gasser (dir.), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann Publishers, p. 205-209, 1988.
- [GEO 90] GEORGEFF M.P., *Readings in Planning*, Morgan Kaufmann Publishers, San Mateo, Californie, 1990.
- [GRO 96] GROSZ B.J., KRAUS S., « Collaborative plans for complex group action », *Artificial Intelligence*, 86, p. 269-357, 1996.

- [HEW 91] HEWITT C., « Open information systems semantics for Distributed Artificial Intelligence », *Artificial Intelligence*, vol. 47, p. 79-106, 1991.
- [HUB 96] HUBER M.J., DURFEE E.D., « An initial assessment of plan- recognition-based coordination for multi-agents teams », *Proceedings of the Second International Conf. on Multi-agents Systems (ICMAS-96)*, p. 126-133, 1996.
- [ISH 92] ISHIDA T., GASSER L., YOKOO M., « Organization self-design of distributed production systems », *IEEE Trans on Knowl and Data Sys DKE4*, vol. 2, n° 123-134, 1992.
- [JEN 91] JENSEN K., ROZENBERG G. (dir.), *High Level Petri Nets, Theory and Applications*, Springer-Verlag, 1991.
- [KAB 95] KABANZA F., « Synchronizing multiagent plans using temporal logic specifications », dans *Proceedings of the First International Conf. on Multi-agents Systems (ICMAS-95)*, p. 217-224, 1995.
- [KET 94] KETCHPEL S., « Forming coalition in the face of uncertain rewards », dans *Proc. of the 12th National Conference on Artificial Intelligence*, vol. 1, p. 414-419, Menlo Park, CA, AAAI Press, 1994.
- [KIN 92] KINNY D., LJUNGBERG M., RAO A., SONENBERG E., TIDHAR G., WERNER E., *Planned Team Activity*, Preproceedings of the Fourth European Workshop on Modeling Autonomous Agents in a MultiAgent World, juillet 1992.
- [KOR 81] KORNFELD W.A., HEWITT C., « The Scientific Community Metaphor », *IEEE Transactions on system, Man and Cybernetics*, vol. 11, n° 1, p. 24-33, janvier 1981.
- [KRE 90] KREPS D.M., *A course in microeconomic theory*, Princeton University Press, 1990.
- [LES 81] LESSER V.R., CORKILL D.D., « Functionally accurate, cooperative distributed systems », *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-11, (1), p. 81-96, 1981.
- [LES 91] LESSER V.R., « A retrospective view of FA/C distributed problem solving », dans *IEEE transaction on Systems, Man and Cybrnetics, Special issue on Distributed AI*, vol. 21, n° 6, p. 1347-1362, 1991.
- [MOR 96] MORAÏTIS P., TSOUKIÀS A., « A Multicriteria Approach for Distributed Planning and Conflict Resolution for Multi-agents Systems », dans *Proceedings of the International Conference on Multi-Agents Systems (ICMAS'96)*, p. 212-219 1996.
- [MOR 99] MORAÏTIS P., TSOUKIÀS A., « Dynamic Planning Model for Agent's Preferences Satisfaction: First Results », dans *Proceedings of the 1st Asia-Pacific Conference on Intelligent Agent Technology (IAT'99)*, Hong Kong, 1999.
- [MÜL 96] MÜLLER H.J., « Negotiation Principles », dans O'Hare et Jennings (dir.), *Foundations of Distributed Artificial Intelligence*, p. 211-229, 1996.
- [NAS 50] NASH J., « The bargaining problem », *Econometria*, n° 18, p. 155-162, 1950.
- [OSS 99] OSSOWSKI S., *Co-ordination in Artificial Agent Societies*, Springer-Verlag, LNAI 1535, 1999.

- [RAO 91] RAO A.S., GEORGEFF M.P., « Modeling Rational Agents within a BDI-Architecture », dans *the Proc. of KR'91 (International Conference on Principles of Knowledge Representation and Reasoning)*, 1991.
- [ROS 82] ROSENSCHEIN J.S., « Synchronization of Multi-agents Plans », dans *Proceedings of AAAI'82*, p. 115-119, 1982.
- [ROS 94] ROSENSCHEIN J., ZLOTKIN G., « Rules of Encounter – Designing Conventions for automated negotiation », *AI magazine*, vol. 15, n° 3, p. 29-46, 1994.
- [SEN 00] SEN S., SARATHI DUTTA P., « Searching for optimal coalition structures », dans *Proc. of ICMAS'2000*, IEEE Press, Boston, Massachusetts, 2000.
- [SHE 95] SHEORY O., KRAUS S., « Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents », dans *Proc. of ICMAS'95*, MIT Press, 1995.
- [SIC 93] SICHMAN J., CONTE R., DEMAZEAU Y., CASTELFRANCHI C., « Reasoning about Others Using Dependence Networks », dans *Proceedings of the 3rd Italian Workshop on Distributed Artificial Intelligence*, AI*IA, Rome, Italie, 1993.
- [SIC 94] SICHMAN J.S., CONTE R., DEMAZEAU Y., CASTELFRANCHI C., « A Social Reasoning Mechanism Based On Dependence Networks », *ECAI 1994*, p. 188-192, 1994.
- [SMI 80] SMITH R.G., « The Contract Net Protocol: High-Level Communication and Control in Distributed Problem Solver », *IEEE Trans. on computers*, vol. 29, n° 12, p. 1104-1113, 1980.
- [SYC 91] SYCARA K., ROTH S., SADEH N., FOX M., « Distributed constrained heuristic search », *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-21(6), p. 1446-1461, 1991.
- [SYC 89] SYCARA K., « Argumentation: Planning Other Agents' Plans », dans *Proceedings of IJCAI'86*, Detroit, Michigan, p. 517-523, 1989.
- [VAU 00] VAUVERT G., EL FALLAH SEGHRUCHNI A., « Coalition formation for egoistic agents », dans *Proceeding of MAMA'2000 (Multi-Agents and Mobile Agents in Virtual Organizations and E-Commerce)*, Wollongong, Australie, 2000.
- [VAU 01] VAUVERT G., EL FALLAH SEGHRUCHNI A., « Coalition formation among strong autonomous and weak rational agents », dans *Proceeding of MAAMAW'2001 (Modeling Autonomous Agents in Multi-Agents World)*, Annecy, France, 2001.
- [VMA 92] VON MARTIAL F., *Coordinating Plans of Autonomous Agents*, Springer-Verlag, LNAI, 1992.
- [ZLO 91] ZLOTKIN G., ROSENSCHEIN J.S., « Cooperation and Conflict Resolution via Negotiation Among Autonomous Agents in Noncooperative Domains », *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, n° 6, p. 1317-1324, novembre-décembre 1991.
- [ZLO 94] ZLOTKIN G., ROSENSCHEIN J.S., « Coalition, cryptography, and stability: mechanisms for coalition formation in task oriented domain », dans *Proc. of the 12th National Conference on Artificial Intelligence*, vol. 1, p. 432-437, Menlo Park, CA, AAAI Press, 1994.
- [ZLO 96] ZLOTKIN G., ROSENSCHEIN J.S., « Mechanism design for automated negotiation and its application to task oriented domains », *Artificial Intelligence*, 86, p. 195-244, 1996.

Chapitre 5

Environnements de développement

5.1. Introduction

Cette dernière décennie a vu fleurir un très grand nombre de modèles, de langages et d'architectures dont le but est de développer des systèmes à plusieurs entités autonomes pouvant coopérer entre elles. Ces entités sont appelées *agents*. Cette prolifération est le signe du grand intérêt pour l'agent et le système multi-agent. Cependant, dans de nombreux cas, les propositions sont conceptuelles et ne sont pas étayées par des réalisations pratiques permettant de les valider. La difficulté de réalisation des systèmes multi-agents est notamment due à la multiplicité, à la richesse et à la complexité des concepts sous-jacents (coordination, interaction, organisation, etc.). Cette complexité rend, par ailleurs, l'utilisation de la majorité des outils existants très difficile par les « non-auteurs » et quasiment impossible par des « non-spécialistes » des systèmes multi-agents.

Ce nouveau paradigme connaît un très vif succès chez les industriels pour qui il promet des outils de conception/implémentation flexibles et adaptatifs. Beaucoup d'industriels sont convaincus que les systèmes multi-agents offrent la solution à plusieurs problèmes de conception, de modélisation et de réalisation que l'on n'a pas su résoudre avec les paradigmes et méthodes existants tels que les objets. Par exemple, pour concevoir le modèle organisationnel d'un système d'information, il semble évident que l'utilisation d'une organisation d'agents faciliterait la conception d'une part et améliorerait le modèle d'autre part.

Les environnements de développement multi-agents sont nécessaires pour couronner le grand succès de ce paradigme. Ces environnements permettent aux utilisateurs de concevoir et de réaliser leurs applications sans perdre de temps à comprendre les différents concepts. D'autre part, une mise en œuvre reste la meilleure validation possible.

Ce chapitre a pour but de faire le point sur les différents environnements de développement de systèmes multi-agents et montrer l'apport de ces environnements par rapport aux outils classiques (objets, systèmes experts, etc.). Il s'agit, d'une part, de présenter les principales composantes d'un environnement de développement de systèmes multi-agents et la différence avec les différentes techniques existantes, et d'autre part, de cerner les caractéristiques d'un environnement de développement.

5.2. Des défis pour les environnements de développement multi-agents

Les systèmes multi-agents ont été appliqués avec succès dans des domaines d'application tels que le contrôle ou la simulation de processus dynamiques complexes (la robotique, le monitoring de procédés industriels ou la simulation d'écosystèmes). Ils offrent une solution aux limites des techniques classiques notamment celles de l'intelligence artificielle.

Les systèmes multi-agents constituent un outil pour modéliser ces domaines complexes mais ils ont également l'ambition de faciliter les différentes étapes de développement. Ils fournissent, en effet, de nombreuses facilités pour développer des systèmes complexes. Ceci est principalement dû à leurs caractéristiques. Cependant, la mise en œuvre d'un système possédant de telles caractéristiques est souvent une tâche complexe. Elle reflète la complexité de l'opérationnalisation des systèmes multi-agents. Le processus de développement est rendu complexe par le fait que les systèmes multi-agents sont :

- des systèmes ouverts : l'idée sous-jacente est d'apporter à l'utilisateur plus de souplesse dans le développement et l'exploitation des systèmes multi-agents. L'utilisateur devrait, par exemple, pouvoir créer une partie de son système, la tester et ensuite développer les autres parties. Une plus grande souplesse est ainsi atteinte lorsque plusieurs parties ou applications développées indépendamment peuvent être connectées ou intégrées à d'autres applications ;

- des systèmes hétérogènes : l'hétérogénéité est une caractéristique très importante des systèmes complexes. Ceci est notamment dû à la nécessité d'accomplir un haut niveau d'interopérabilité entre des agents hétérogènes. Différents types d'hétérogénéité peuvent être considérés :

- multiplicité des langages d'implémentation,
- multiplicité des plates-formes d'exécution,
- multiplicité des formalismes de représentation des connaissances ;

Le processus de développement d'un système multi-agent devrait ainsi tenir compte du contexte et des compétences du développeur. Cet utilisateur souhaiterait avoir la possibilité d'utiliser son langage, sa plate-forme, etc. Il souhaiterait également pouvoir réutiliser ses programmes comme briques de base pour la construction de son système ;

- des systèmes en interaction : chaque agent est conçu pour interagir correctement (c'est-à-dire en respectant un protocole) avec les autres agents du système. Un environnement de développement guide et oriente le développeur pour utiliser les protocoles prédéfinis. Il devrait également offrir une bonne documentation de ces différents protocoles ;

- des systèmes distribués : un système multi-agent est par définition un système distribué. Ceci permet à ces systèmes d'hériter de tous les avantages des systèmes distribués mais également de la complexité de leur conception.

La conception et la réalisation des systèmes multi-agents est ainsi difficile. Ces systèmes ont tous les problèmes des systèmes distribués et concurrents. D'autres problèmes sont liés à leurs caractéristiques et sophistications telles que la flexibilité des interactions et la dynamique des organisations.

Les caractéristiques des systèmes multi-agents engendrent, en effet, des obstacles qu'il faudrait éliminer ou contourner pour les opérationnaliser. Par exemple, les deux fonctionnalités suivantes sont fortement contraintes par les caractéristiques précédentes telles que la distribution et l'hétérogénéité.

Observation et contrôle du système : la construction d'un système multi-agent nécessite l'identification des différents agents qui peuvent coopérer pour résoudre un problème. Un des problèmes que pose la conception et la réalisation des systèmes multi-agents est de savoir comment observer, comprendre et contrôler la coordination dans une organisation d'agents. Les systèmes multi-agents ont plusieurs caractéristiques qui rendent très complexes leur observation et leur contrôle. Le problème majeur est la distribution. Chaque agent a son propre ensemble de variables représentant ses données et ses états mentaux, et la synchronisation est limitée uniquement à ce qui peut se faire par envoi de messages. Ces facteurs ne facilitent pas l'obtention ou l'observation de l'état du système multi-agent. Cet état n'est qu'un ensemble d'états locaux distribués. Ce problème se cache derrière plusieurs problèmes de développement de systèmes multi-agents. Par exemple, pour introduire un nouvel agent dans une application, le développeur souhaite souvent comprendre ce qui existe ou, en d'autres termes, l'application à laquelle le nouvel agent serait intégré.

Exécution et *debugging* : il est difficile de suivre, par exemple, l'exécution d'un agent pas à pas et comprendre son exécution. Ceci est notamment dû à ses interactions avec le monde extérieur et les autres agents. Un agent peut ainsi

interrompre son processus de délibération pour répondre à la requête d'un autre agent de son organisation.

5.3. Vers une généralité de l'opérationnalisation des SMA

De nombreuses notions entrent en jeu dans la conception d'un SMA, comme la description des agents à travers leurs comportements, leurs connaissances ou la description des interactions. Ces concepts donnent lieu à de nombreux modèles spécifiques utilisant des formalismes variés. Selon qu'ils s'inscrivent dans une logique de décomposition de problèmes ou d'intégration de systèmes, les auteurs centrent leurs travaux plus particulièrement sur les agents ou les interactions. Yoav Shoham introduit la notion de *programmation orientée agent* [SHO 93] (en référence à la programmation orientée objet) en proposant de donner une vision sociétale de la programmation, dans laquelle les unités opérationnelles sont des entités dotées d'un état mental, de compétences et de croyances et en s'attachant à définir précisément les modèles de ces agents. D'autres se préoccupent plutôt de l'interopérabilité entre systèmes et proposent des modèles pour l'interaction et la coordination évoluant vers ce que l'on peut qualifier de *programmation orientée interaction*. Les outils génériques relèvent donc de deux soucis principaux : réaliser les modèles et intégrer ces modèles en des SMA opérationnels.

Ces deux préoccupations conduisent à des outils de nature différente, qui souvent font appel l'un à l'autre. On peut distinguer :

- des outils de base (tels des langages ou les architectures logicielles) qui mettent en œuvre des modèles de base d'agent, d'interaction, d'organisation, etc., mais qui restent souvent centrés sur les modèles eux-mêmes ;
- des outils intégrés qui sont plus centrés sur le SMA résultant et se préoccupent plus de l'assemblage des modèles de base et de l'assistance au développement.

Nous présentons dans cette partie une rétrospective de l'évolution des environnements dédiés à la conception de SMA à travers une description des outils (langages, architectures, environnements intégrés) issus de ce cheminement vers la généralité.

5.3.1. Des langages de programmation

Généralement, pour la réalisation concrète du SMA, les concepteurs des modèles se tournent vers la production de langages d'implantation dédiés à chaque aspect. Mais ils doivent alors nécessairement fournir des constructions et primitives de haut niveau dans les outils d'implantation. On trouve essentiellement deux approches pour fournir ces structures de haut niveau. La première est de spécifier de nouveaux

langages où les caractéristiques agent et multi-agent sont des primitives. La seconde est d'utiliser des langages génériques pour construire des couches de niveau SMA. Les environnements proposés sont dans le premier cas des langages de programmation à part entière, compilés ou interprétés et, dans le deuxième cas, des bibliothèques de primitives et structures de données disponibles pour des langages hôtes.

5.3.1.1. *Un langage d'agent : AGENT0*

AGENT0 est un précurseur dans la mesure où il propose un nouveau style de programmation et un vrai langage de programmation intégrant un grand nombre des concepts définissant la notion d'agent. Shoham présente sa programmation orientée agent [SHO 93] comme une spécialisation de la programmation orientée objet, où les modules (appelés agents) sont munis d'un état mental composé de croyances sur eux-mêmes, sur les autres et sur le monde, de compétences et de capacités de décision. Il propose un langage de programmation interprété AGENT0 permettant de programmer les agents.

AGENT0 est fondé sur un langage formel réduit utilisant la logique modale pour décrire les états mentaux. Les agents sont programmés en termes de règles d'échange mental. AGENT0 a trois composantes : 1) un système logique pour définir les états mentaux de l'agent ; 2) un langage de programmation pour programmer des agents et 3) un processus « agentification », pour compiler les programmes agents en un système exécutable (bas-niveau). Les concepts de base sont :

- les *Actions* provoquant des faits déterminés (externes ou internes) par des décisions,
- les *Capacités* indiquant dans quelle mesure l'agent est capable d'effectuer une action,
- les *Croyances* faisant référence à l'état du monde, aux états mentaux et aux compétences des autres, et contraignant les décisions,
- les *Obligations* traduisant des engagements et proposant des alternatives pour les décisions.

Le moteur de l'agent contrôle l'évolution de l'état mental et les actions sont exécutées en tant qu'effets de bord de cette évolution. Le cycle de contrôle est assez simple (voir figure 5.1). Il consiste en la répétition de la boucle suivante : lire les messages, mettre à jour l'état mental et exécuter les actions prévues pour le cycle courant.

AGENT0 est défini par une grammaire que l'on peut trouver dans [SHO 93]. Il est écrit en Common LISP mais suit une organisation en règles de production. Il permet d'énoncer :

- des faits (expressions de prédicats),
- des actions privées ou de communication (conditionnelles ou non), dont voici quelques exemples :
 - DO <temps> <action privée> : exécute une action interne à l'agent,
 - INFORM <temps> <agent> <fait> : communique un fait à un autre agent,
 - REQUEST <temps> <agent> <fait> : requiert l'exécution d'une action à un autre agent ;
- des règles d'exécution des engagements exprimées par l'instruction COMMIT <condition sur les messages> <condition sur l'état mental> <liste d'actions>.

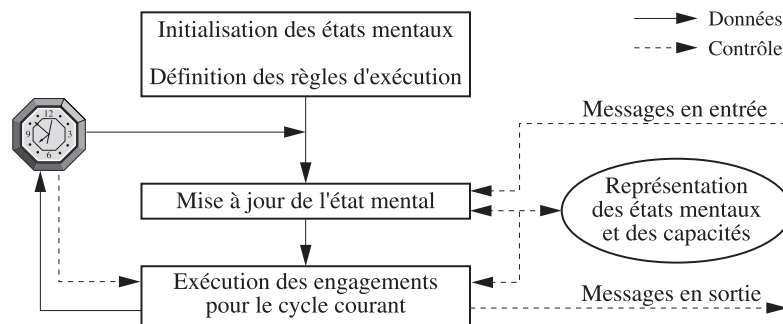


Figure 5.1. Le cycle de contrôle d'un agent AGENT0

Un programme AGENT0 peut être vu comme une séquence de règles de production exprimant le comportement de l'agent, précédées par une définition des capacités de l'agent et de ses croyances initiales. La dynamique de l'agent est réalisée par un déclenchement en chaînage avant de ces règles.

5.3.1.2. Panorama des langages multi-agents

Nous distinguons deux types de langages : des langages à part entière et des bibliothèques pour des langages traditionnels.

5.3.1.2.1. Des langages à part entière

Les productions de langages d'agents à part entière sont très diversifiées. Elles étendent les bases jetées par AGENT0, soit du point de vue des modèles mis en œuvre, soit du point de vue des outils résultants.

LALO [GAU 95] reprend les concepts de base de AGENT0, mais il en fournit une expression plus large du comportement des agents. L'utilisateur peut choisir de

spécifier le comportement à l'aide de règles comme dans AGENT0, à l'aide d'un ensemble de plans ou plus simplement d'une façon procédurale. L'intérêt de LALO par rapport à AGENT0 réside surtout dans son aspect compilé. Le compilateur LALO génère à partir du programme de l'agent, un code source C++ pour l'agent, et le code source d'un programme d'installation dans la base de données (orientée objet) des agents qui contient au lancement les croyances et engagements initiaux de l'agent (voir figure 5.2). Grâce à cette base, LALO constitue une avancée pratique par rapport à AGENT0.

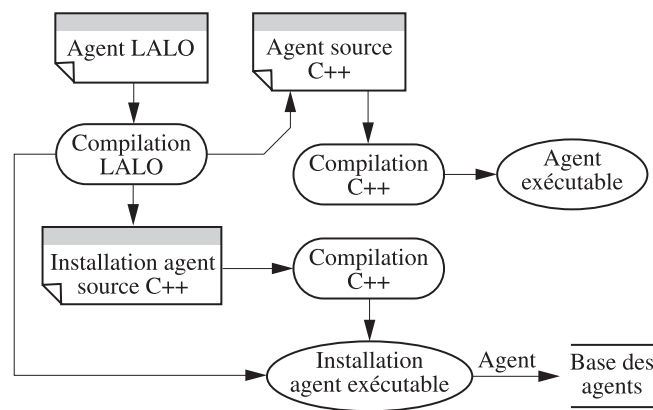


Figure 5.2. Le compilateur LALO

PLACA (*PLAnning Communicating Agent*) [THO 95] est une version raffinée de AGENT0. Ce dernier est seulement un prototype pour illustrer les principes de la programmation par agents. PLACA rajoute aux agents la capacité de planifier, et de communiquer des requêtes pour des actions à travers des buts de haut niveau. PLACA reste, cependant, un langage expérimental.

Dans PLACA et AGENT0, la relation entre le système logique qui définit les états mentaux de l'agent et le langage de programmation interprété est vaguement définie. Pour apporter une solution à ce problème M. Fisher a réalisé METATEM [FIS 94]. Ce dernier est utilisé pour définir un ensemble d'agents qui peuvent être activés de façon concurrente. Chaque agent a la capacité de communiquer avec les autres agents par envoi de messages asynchrones. Il est programmé en donnant une spécification en logique temporelle de son comportement.

Même s'ils ne permettent pas de développer complètement un système d'agents, il est intéressant de souligner les efforts fournis pour proposer des langages

génériques pour l'interaction et de se pencher brièvement sur leur opérationnalisation. Du point de vue de leur opérationnalisation, deux approches se retrouvent : d'une part, la recherche d'un standard d'interaction indépendant des agents, d'autre part, la construction de bibliothèques adaptées à des agents particuliers. On peut assimiler ces démarches à une externalisation de l'opérationnalisation des fonctions d'interaction des agents.

KQML est le résultat d'un effort de recherche visant l'interopérabilité entre agents logiciels. Son principal attrait pour le développement des SMA est son souci de découpler l'interaction de la communication. C'est à chaque implémentation de choisir les primitives retenues parmi l'ensemble proposé par le langage. Si par sa syntaxe parenthésée KQML peut se rapprocher d'un langage de programmation comme LISP, il constitue plutôt un cadre conceptuel commun aux agents pour interagir. Il existe de nombreuses expérimentations implantées à l'aide de langages de programmation divers (LISP, C) et même une version utilisant SQL [CHU 94].

Le langage IL [DEM 95] permet l'utilisation de protocoles et introduit une composante multi-agent dans le langage. Son environnement d'exploitation propose un mécanisme indépendant et générique d'enregistrement des agents et de routage des messages. IL est implanté sous la forme de bibliothèques d'exploitation disponibles en JAVA et C++.

Le consortium FIPA, enfin, propose comme standard le langage FIPA ACL s'inspirant de KQML mais introduisant l'exploitation des protocoles comme le fait IL. Son originalité est de proposer pour l'implantation de s'appuyer sur les primitives CORBA.

5.3.1.2.2. Des bibliothèques pour des langages traditionnels

De nombreuses expériences ont été menées afin non pas de créer des langages originaux purement SMA mais de fournir des bibliothèques SMA aux langages existants. On pourra remarquer que tous les champs de la programmation traditionnelle auront été explorés.

Le domaine des langages logiques a été très utilisé pour formaliser les SMA [WOO 95]. Les formalismes logiques sont bien adaptés à la modélisation, mais posent des difficultés d'implémentation [SHO 93].

De très nombreux travaux sur les langages SMA sont implantés comme extension de langages fonctionnels. On pourra citer, par exemple, les bibliothèques de coordination AgenTalk [KUW 95]. AgenTalk propose la construction de protocoles sous la forme d'automates à états finis décrits sous forme de scripts en Common LISP. Ces protocoles peuvent être définis incrémentalement par héritage.

Des fonctions de manipulation sont fournies permettant de manipuler ces scripts dans les agents ; elles sont disponibles en Common LISP ou C++.

Quelques approches proposent d'utiliser des spécifications formelles établies à l'aide de langages formels tels Z pour concevoir des SMA. Luck et D'Inverno proposent notamment l'outil Agentis. DESIRE [BRA 95] propose un langage formel de spécification de composants pour la construction de SMA.

Il serait impossible d'être exhaustif dans l'établissement d'une liste de bibliothèques SMA en langages objets. En particulier, de très nombreux travaux sont menés notamment sous l'influence du succès des agents mobiles pour le *web* en Java. Ils donnent lieu à de très nombreux outils académiques ou commerciaux, tant sur le plan des agents que des interactions.

De nombreuses réimplantations de systèmes issus de la programmation fonctionnelle vers C++ ou Java [DIN 98] montrent bien l'engouement des concepteurs de SMA pour les objets. La programmation par objets concurrente est aussi un outil très naturel pour implémenter les agents. Les langages d'acteurs ou les objets concurrents fournissent traditionnellement un cadre puissant et très populaire d'implémentation de SMA. MERINGIV [FER 91] ou REACTALK [GIR 93] (basé sur AcTalk [BRI 89]) sont de bons exemples d'outils acteurs étendus vers les agents. [JEN 95] propose une extension de la programmation Prolog parallèle (*Stream Logic Programing*) pour programmer des agents profitant des aspects acteurs et Prolog. L'acteur, élément de base de la programmation par objets concurrente, peut être considéré utilement comme une brique de base pour l'implémentation de modèles d'agents. Le principal avantage du concept d'acteur est qu'il s'intègre harmonieusement avec le paradigme d'objet. Cette fructueuse intégration permet de bénéficier de tous les développements conceptuels, méthodologiques et technologiques liés à la programmation par objets.

5.3.1.3. Discussion

Tous ces langages montrent, cependant, de sérieuses limitations sur un certain nombre de points. Ils sont la plupart du temps limités au type de modèle d'agent particulier qu'ils développent. S'ils sont « centrés agents », on constate en termes de modèles une grande focalisation sur les aspects de décision et de grandes limitations sur les aspects sociaux. On peut remarquer que les langages portent très majoritairement sur des approches BDI. On constate aussi la quasi-absence de langages génériques pour construire des SMA réactifs alors que de nombreuses bibliothèques à base d'acteurs (bien adaptés pour les réactifs) sont disponibles.

Les langages d'interaction sont le plus souvent définis indépendamment des agents alors que les bibliothèques d'interaction sont très souvent dédiées à des types d'agents particuliers. On se trouve face à un dilemme : soit laisser totalement libre le

modèle d'agent, et à ce moment-là garder un niveau d'abstraction très élevé et peu opérationnel pour le langage (comme KQML), soit fournir un langage opérationnel mais qui contraint fortement le modèle et la programmation des agents (comme AgenTalk). Pour remédier à ces lacunes des essais d'intégration multilangages SMA ont été tentés comme une intégration AGENT0 et KQML dans Agent-K [DAV 94].

Toutes ces remarques montrent que finalement l'approche langage de programmation SMA tout en laissant toutes les responsabilités au programmeur manque de souplesse et de généricité. Les langages ne peuvent être facilement étendus pour aborder d'autres problèmes ou d'autres domaines d'application.

Afin de faire un autre pas vers une opérationnalisation plus accessible des systèmes multi-agents, des travaux ont tenté de réutiliser des paradigmes et langages existants afin de définir des architectures d'agents. L'objectif est d'avoir des outils plus génériques qui permettent d'une part de développer des applications plus simplement et plus rapidement et d'autre part d'aborder d'autres problèmes que posent la conception et la réalisation de ces systèmes. Nous tentons de faire une synthèse de ces différents travaux dans le paragraphe suivant.

5.3.2. Des architectures logicielles

Des travaux tentent de construire leur SMA autour d'architectures logicielles classiques plus évoluées qui permettent une organisation et une programmation plus aisée des capacités des agents. Nous ne nous intéressons pas ici aux modèles logiques des agents, mais à leur expression à l'aide de structures logicielles consacrées. Pour l'étude des modèles, on pourra se reporter au chapitre 2. La majorité de ces travaux est issue de l'intelligence artificielle. Les systèmes multi-agents sont dans ce cas une solution que les auteurs ont trouvée pour étendre les champs d'application et faire face à la complexité de ces applications. Nous n'avons retenu que les types d'architectures qui apportent réellement de la généricité.

5.3.2.1. Panorama des architectures logicielles génériques utilisées en SMA

Certaines architectures remettent au goût du jour des architectures consacrées de l'intelligence artificielle.

5.3.2.1.1. Des architectures à base de tableaux noirs

L'architecture de tableau noir, après avoir servi à implémenter des SMA [FER 95, LAA 88] est aujourd'hui un moyen efficace pour implémenter des architectures internes d'agents.

L'architecture ASTRO (agent spécialiste pour le temps réel et la coopération) [OCC 98], par exemple, propose l'intégration de capacités cognitives et réactives par l'exploitation du parallélisme dans la structure même de l'agent. Le découplage entre les tâches d'évaluation/raisonnement et de perception/communication permet une supervision continue de l'évolution de l'environnement.

Dans le modèle opérationnel d'ASTRO, les différentes capacités nécessaires sont organisées en modules asynchrones d'un système tableau noir parallèle. Le modèle du monde est stocké dans le tableau noir du domaine. Tous les modules sont gérés par l'unité de contrôle. Cette approche multimodule permet une description indépendante et modulaire de chacune des tâches de perception et d'action dans des modules séparés.

L'approche tableau noir pour construire les architectures internes est devenue classique ; on peut citer l'architecture GTMAS [CHE 93] ou l'architecture Guardian [HAY 95], agent intelligent adaptatif qui fait le suivi en temps réel d'un patient dans une unité de réanimation. La mise en réseau des architectures à base de tableaux noirs, introduite dans DVMT [DUR 88], offre les potentialités requises pour la construction du SMA très bien adaptées à des agents peu nombreux de grosse granularité [OCC 93].

5.3.2.1.2. Les architectures à base de règles

Les systèmes de production héritent des concepts des systèmes experts. Ils proposent des outils de représentation de la connaissance et du raisonnement qui sont particulièrement bien adaptés à la construction des agents. On a pu voir que le moteur d'agent de AGENT0 se fondait déjà sans le dire sur ces concepts. On peut citer dans cette catégorie l'architecture d'agent utilisée dans SYNERGIC [CAR 95]. Les agents sont des modules de connaissances exprimées de façon déclarative sous la forme de règles de production. Dans Pengi [AGR 87], un agent réactif est représenté sous la forme d'une base de règles. La partie condition de la règle représente le stimulus et la partie action représente la réponse appropriée.

5.3.2.1.3. Des architectures multiparadigmes

De nombreux travaux tentent de faire cohabiter ou de fusionner différentes techniques au sein d'architectures multiparadigmes afin de répondre à des besoins particuliers ou d'apporter plus de flexibilité à leurs systèmes.

Tout comme l'approche tableau noir, l'objectif des architectures en couches, dont Interrap est un bon représentant [MUL 94], est de fournir un cadre d'organisation individuelle des compétences des agents. Si les tableaux noirs proposent un découpage que l'on peut qualifier de *fonctionnel* (tous les modules sont considérés au même niveau), les architectures en couches préconisent un découpage

hiérarchique permettant d'intégrer des capacités de nature différentes. Les capacités sont réparties en trois niveaux. On retrouve sur chaque niveau des capacités de perception, de décision et de raisonnement. Le niveau inférieur est basé sur des comportements réactifs de base. Dans Interrap, les capacités de raisonnement de ce niveau sont réduites à des déclenchements de procédures exécutables en fonction de l'évaluation des perceptions. Le second niveau est chargé de la planification locale, il est sollicité si le premier niveau détecte une situation qu'il n'est pas capable de traiter de façon réactive. Le dernier niveau concerne la planification coopérative afin de gérer le traitement des tâches collectives.

Dans le même esprit, on pourra se référer aussi aux architectures Touring machines de Fergusson [FER 92] et ASIC de Boissier [BOI 96].

L'architecture de *subsumption* introduite par R. Brooks [BRO 86] construit les capacités des agents par empilement vertical de modules responsables de comportements très limités soumis les uns aux autres. Une couche peut être par exemple composée d'un réseau d'automates. Chaque automate prend en entrée un signal et génère un autre signal. Les signaux d'entrée représentent les stimuli (délivrés par les capteurs). Les signaux de sortie sont transmis aux effecteurs ou à d'autres automates.

Ce type d'architecture fixe un cadre d'exécution par la distribution des compétences sur les niveaux, un contrôle hiérarchique entre les niveaux ainsi que les modalités de communication entre modules.

L'architecture d'agents DIMA [GUE 99] propose de décomposer chaque agent en différents modules dont le but d'intégrer des paradigmes existants, notamment des paradigmes d'intelligence artificielle.

5.3.2.2. Discussion

Ces architectures sont employées communément aujourd'hui pour les organisations internes des agents. Même si les architectures en couches sont mieux adaptées à l'intégration des aspects réactifs et cognitifs, le choix de l'architecture est souvent indépendant du modèle qu'elle met en œuvre. Il s'agit bien des caractères génériques des architectures qui poussent les concepteurs à requérir leurs services.

Ces architectures présentent en effet des qualités indéniables de génie logiciel. Leur organisation modulaire permet entre autre l'intégration de capacités très hétérogènes en termes de programmation. Ceci est particulièrement le cas pour les architectures à tableau noir et en couches. Mais des aménagements apportés aux systèmes à base de règles comme les « agents programmes » (boîtes noires) de SYNERGIC permettent d'intégrer des éléments hétérogènes. Cette hétérogénéité est nécessaire pour répondre à l'intégration des tâches très diversifiées que doit remplir

un agent, depuis le raisonnement jusqu'à l'interaction, qui peuvent faire appel à des concepts de système des plus spécialisés. L'intérêt est de pouvoir réutiliser des méthodes existantes, notamment des méthodes de l'intelligence artificielle, comme dans DIMA.

De même, et en écho à l'hétérogénéité, la modularité introduit beaucoup de souplesse dans la gestion des capacités en autorisant à moindre frais l'échange de modules dans un but d'évolutivité ou de test. L'architecture peut faire de l'agent un système ouvert.

La modularité offre en outre une particulière adaptation à l'exécution parallèle des différentes compétences des agents. Les agents du SMA sont naturellement distribués, mais la nature même des tâches fait que les agents sont eux-mêmes fortement parallélisables. Les travaux sur le parallélisme dans les tableaux noirs sont par exemple très nombreux [DEC 90, HAY 95, OCC 98].

Un trait d'union entre ces architectures réside dans la généralité de leurs structures de contrôle. Munies de moteurs d'inférences ou d'unité de contrôle souvent indépendants du problème, les architectures déchargent l'utilisateur du contrôle de la résolution du problème.

Au cours de la conception, les architectures fixent *a priori* un certain nombre de choix d'implémentation et la façon de programmer. Des cadres précis sont fournis pour la représentation des données ou pour la programmation des capacités. En particulier, ces architectures mélangent presque toujours les aspects déclaratifs et aspects opératoires.

Il semble que l'adoption d'architectures logicielles évoluées témoigne d'un plus grand souci de généralité et est plus propice à l'évolution de ces systèmes dédiés vers des systèmes de développement intégrés génériques. Cependant, elles restent difficilement utilisables par un utilisateur « spécialiste SMA » qui n'a pas conçu ou développé l'architecture et souvent inutilisables par les autres utilisateurs. La notion d'architecture a le mérite d'avoir initié un effort important vers la notion de conception de SMA. Mais il apparaît clairement que le cycle de développement des SMA dépasse largement l'étape de réalisation. Le concepteur a besoin de bien d'autres outils pour mener à bien sa tâche, pour l'aider dans les différents stades du développement. Ce sont de réels environnements de développement qui se construisent aujourd'hui autour des architectures et des langages.

5.3.3. Des environnements de développement

Après avoir acquis une bonne expérience dans le développement des systèmes multi-agents et les systèmes mono-agents, les travaux sur les architectures d'agents

ont tenté d'identifier les différentes fonctions et composantes (perception, communication, délibération, etc.) d'un agent et la façon dont elles sont organisées pour fournir le comportement global d'un agent. Ces différentes composantes ont ensuite été réifiées pour faciliter la spécification d'un agent.

Les architectures d'agents offrent une solution à une partie des problèmes que pose le développement des applications multi-agents. Cependant, elles ne résolvent pas le problème de la difficulté d'implémenter les SMA et celui de leur non-réutilisabilité notamment par un non-spécialiste des SMA. Une architecture est donc nécessaire, mais elle n'est pas suffisante pour offrir au développeur toutes les facilités requises. Plusieurs environnements ont ainsi été proposés en vue d'offrir au développeur des outils pour analyser, concevoir, implémenter et tester des systèmes multi-agents. Le paragraphe suivant présente une classification de ces différents environnements.

Panorama des environnements de développement multi-agents

Le processus de développement d'un système multi-agent est similaire au processus des logiciels classiques. Il repose sur les étapes suivantes : analyse, conception, implémentation et validation.

La complexité de chacune de ces étapes est liée au type du système multi-agent développé. Par exemple, l'implémentation d'agents réactifs n'est pas aussi complexe que l'implémentation d'agents cognitifs. En revanche, les agents des systèmes cognitifs correspondent souvent aux entités du système modélisé, l'analyse de ces systèmes est, dans la majorité des cas, moins compliquée que celle des systèmes réactifs. D'autre part, il est plus simple de tester/déboguer une simulation qu'une application où les agents sont réellement distribués.

La comparaison des différents environnements est donc un problème difficile. Plusieurs travaux l'ont abordée mais, à l'image de ces environnements, les différentes solutions ne sont pas complètes. Le but de ce paragraphe n'est pas de comparer les différents environnements. Il ne s'agit pas non plus de trouver une meilleure classification. Le but de la classification que nous proposons est de mieux présenter les environnements existants. Nous nous sommes, en effet, basés sur les différentes étapes du processus de développement mises en œuvre et sur le type d'applications visées. Les environnements peuvent ainsi être regroupés en cinq catégories (voir tableau 4.1).

Les outils pour la simulation : leur principal intérêt est de fournir un ensemble d'outils et de bibliothèques pour faciliter le développement de simulations multi-agents. La majorité de ces outils intègre un noyau de simulation à événements discrets. Ces noyaux de simulation permettent :

- d’observer, de comprendre et de contrôler l’évolution des différents agents,
- de simuler l’environnement dans lequel ces agents évoluent. Un modèle de simulation à événements discrets peut être utilisé pour simuler l’évolution temporelle des différentes entités de l’environnement.

Catégorie	Environnements
Outils pour la simulation	Swarm [MIN 96], Cormas [BOU 98], Geamas [MAR 97], Mice [DUR 88]
Outils pour l’implémentation	Actalk [BRI 89], DECAF [GRA 00], JACK [BUS 00], MAGES IV [BOU 92], Magique [BEN 97], Mocah [ABC 99], OSACA [SHE 96], Pandora II [MAR 89]
Outils pour la conception	Alaadin [FER 98], JAF [VIN 01], Maleva [LHU 98]
Outils pour la conception et l’implémentation	AgentBuilder [AGE 99], DIMA [GUE 99], Mask [OCC 98], Mercure [BRI 00], MACE [GAS 87]
Outils pour la conception, l’implémentation et la validation	Desire [BRA 95], dMars [DIV 98], MAST [HAN 99], Zeus [NWA 99]

Tableau 4.1. Classification des environnements de développement

Les outils pour l’implémentation implémentent des architectures d’agents. Ces architectures sont conçues et implémentées dans le but de faciliter le développement des agents (voir paragraphe 5.3.2). Les architectures existantes varient des simples objets actifs (par exemple : Actalk) ou des objets actifs dotés d’un langage de communication tel que KQML à des architectures plus complexes basées sur des propositions conceptuelles telles que les états mentaux.

Les outils pour la conception sont en majorité basés (voir, par exemple, JAF et Maleva) sur un modèle componentiel. Ils montrent l’intérêt de l’utilisation des composants logiciels pour la conception des systèmes multi-agents et proposent une démarche pour la conception de ces systèmes. Par exemple dans MALEVA, chaque agent peut être conçu comme une association de composants de base (simples) ou de composants composites. Ceci permet de définir des agents de différentes granularités. D’autre part, un système multi-agent est conçu comme un ensemble de composants représentant ces différents agents.

Les outils pour la conception et l'implémentation offrent un ensemble d'utilitaires pour guider la définition d'un groupe d'agents ou l'introduction d'un nouvel agent. Par exemple, JACK et DECAF sont des *frameworks* d'agents. JACK est basé sur l'architecture BDI et DECAF est basé sur une architecture modulaire. Ces outils fournissent également un ensemble de facilités de développement indépendantes de l'architecture. JACK est fondé sur une méthodologie qui repose sur deux types de modèles :

- les modèles externes regroupant le modèle d'agents et le modèle d'interactions. Ce dernier modèle décrit les relations statiques entre agents ;
- le modèle interne est basé sur le modèle BDI. Il permet de spécifier les différents états mentaux d'un agent.

Les outils pour la conception, l'implémentation et la validation, en plus de la conception et l'implémentation, proposent des outils pour la validation. A l'image des travaux théoriques sur ce thème, ils sont très peu nombreux. Par exemple, MAST fournit des outils pour concevoir et implémenter des applications multi-agents et une infrastructure d'exécution répartie et de test des applications multi-agents développées (validation, test, exploitation).

5.4. Exemples d'environnements de développement

Pour mieux décrire le processus de développement d'un système multi-agent, les paragraphes suivants décrivent quatre environnements : MACE, DIMA, MASK et Zeus. Ce choix n'est pas lié à la classification précédente.

5.4.1. MACE

MACE [GAS 87] est le premier environnement de conception et d'expérimentation de différentes architectures d'agents dans divers domaines d'application. L'objectif était de fournir des supports de haut niveau pour des applications dont les composantes sont de granularités variées.

5.4.1.1. Agents

Dans MACE, un agent est un objet actif qui communique par envoi de messages. Ces agents existent dans un environnement qui regroupe tous les autres agents et toutes les autres entités du système.

Un agent peut effectuer trois types d'actions : changer son état interne, envoyer des messages aux autres agents, envoyer des requêtes au noyau MACE pour contrôler les événements internes (par exemple, le *timer*).

Chaque agent est doté d'un moteur qui représente la partie active de l'agent. Ce moteur détermine l'activité de l'agent et la façon dont les messages sont interprétés.

5.4.1.2. *Outils de développement*

Les principales composantes de MACE sont :

- une collection d'agents liés au domaine d'application ;
- une collection d'agents « système » prédéfinis qui fournissent plusieurs fonctionnalités telles que l'interprétation des commandes, l'interface utilisateur et l'interface avec les autres agents du système, la création d'agents, le traitement d'erreurs, la gestion de l'historique et le contrôle d'exécution ;
- une bibliothèque d'outils et de fonctions que tous les agents peuvent utiliser : un simulateur, plusieurs moteurs d'agents, la gestion des erreurs et le traitement de messages ;
- une base de données, ainsi qu'un agent qui gère les interactions avec cette base de données. Cette dernière est utilisée pour mémoriser les descriptions des différents agents, traiter ces descriptions et éventuellement recréer des agents à partir de ces descriptions.

Le principe de MACE est ainsi analogue au principe de l'environnement Smalltalk dans lequel tout est objet. Les agents ne sont donc pas seulement utilisés pour représenter les entités du domaine d'application, ils représentent également le système d'exploitation, les interfaces, etc.

D'autre part, MACE offre un langage de description d'agents qui peut être utilisé par le concepteur pour définir une nouvelle classe d'agents. L'ensemble des descriptions est stocké dans une base de données.

5.4.1.3. *Applications*

Les auteurs ont constaté que le processus de développement d'un système multi-agent est très complexe. Le premier domaine d'application de MACE est la conception de systèmes multi-agents. Les premiers agents de MACE ont donc été exploités pour faciliter le développement de systèmes multi-agents ainsi que l'utilisation du système par le système multi-agent développé. Ces agents offrent un ensemble de services, une collection de facilités que tous les agents peuvent utiliser y compris l'agent humain qui veut développer son système multi-agent.

MACE a été également utilisé pour développer des simulations d'applications distribuées.

5.4.2. DIMA (développement et implémentation SMA)

DIMA est principalement fondée sur la réutilisation de *frameworks* [GUE 97, GUE 99, GUE 00]. et une architecture d'agents modulaire. La première version de DIMA a été implémentée en Smalltalk-80 et a été ensuite portée en JAVA.

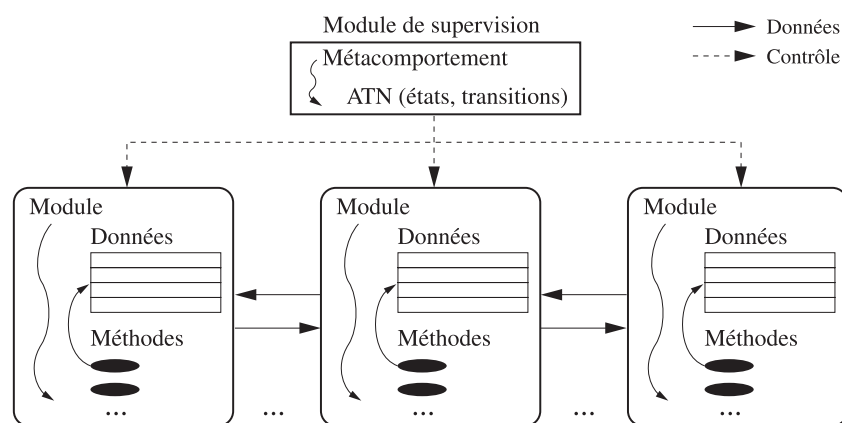


Figure 5.3. Architecture d'agents DIMA

5.4.2.1. Agents

L'architecture d'agents DIMA (voir figure 5.3) propose de décomposer chaque agent en différents modules dans le but d'intégrer des paradigmes existants notamment des paradigmes d'intelligence artificielle. Ces modules représentent les différents comportements d'un agent tels que la perception (interaction agent-environnement), la communication (interaction agent-agent) et la délibération. Un agent peut ainsi avoir un ou plusieurs modules qui peuvent être réactifs ou cognitifs. Un comportement réactif est décrit par un ensemble de méthodes (au sens objets) alors qu'un comportement cognitif peut avoir des méthodes qui activent des mécanismes de raisonnement (par exemple, un moteur d'inférence).

Pour gérer les interactions entre ces différents comportements, DIMA propose un module de supervision représentant le métacomportement de l'agent. Ce métacomportement réifie le mécanisme de contrôle de l'agent, il gère les interactions entre les différents modules et permet à l'agent d'observer ces comportements. Un agent est ainsi une entité proactive et autonome.

Cette architecture permet de dépasser la dichotomie classique en offrant la possibilité de développer des applications multi-agents dont la granularité des agents est variable. Elle offre ainsi la possibilité de concevoir et réaliser différents types d'agents :

- des agents réactifs, décrits par des comportements réactifs,
- des agents cognitifs, décrits par des comportements cognitifs,
- des agents hybrides, conçus pour allier des capacités réactives à des capacités cognitives, ce qui leur permet d'adapter leur comportement en temps réel à l'évolution de leur univers.

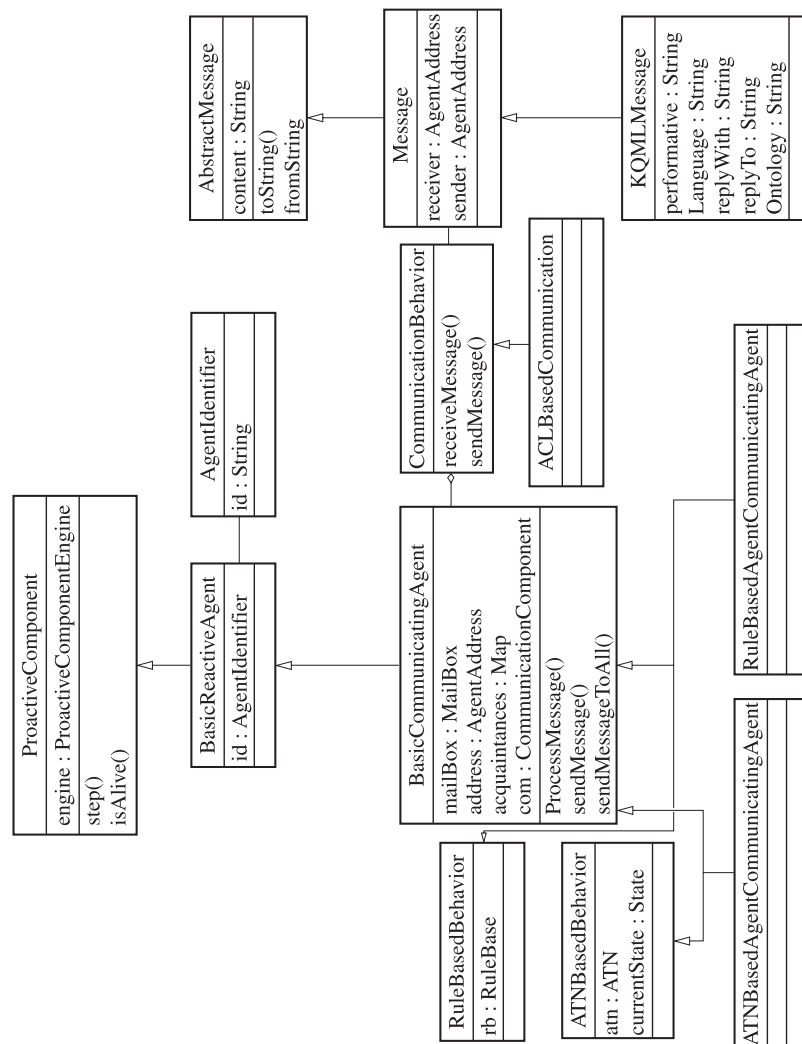


Figure 5.4. Exemple de bibliothèques de DIMA

5.4.2.2. Outils de développement

La principale caractéristique de DIMA est son architecture modulaire et des bibliothèques offrant les briques de base pour construire des modèles d'agents divers. Ces différentes briques ont pour but d'offrir à l'utilisateur, d'une part, une grande variété de paradigmes (par exemple automate, règle, etc.) et, d'autre part, une implémentation des différentes propositions conceptuelles introduites par la communauté multi-agent (BDI, KQML, ACL, etc.). Les bibliothèques de DIMA (voir figure 5.4) regroupent des classes qui peuvent être réutilisées et/ou adaptées pour construire facilement des agents. Ces classes peuvent être instanciées ou sous-classées pour implémenter un comportement. Par exemple, pour la communication, DIMA offre deux classes : *CommunicationBehavior* (basée sur les envois de messages asynchrones) et *ACLBasedCommunication* (basée sur le standard ACL de la FIPA).

L'implémentation d'un SMA consiste en l'implémentation des différents agents et éventuellement l'implémentation des objets représentant leur environnement.

Les principales étapes pour implémenter un système multi-agent sont les suivantes :

- implémentation des agents :
 - détermination du type d'agents, c'est-à-dire le choix de l'ensemble des modules de l'agent ainsi que les classes de la bibliothèque décrivant ces modules,
 - éventuellement, implémentation de la classe décrivant les modules choisis ou spécialisation de classes de la bibliothèque,
 - description de l'ATN décrivant le métacomportement de l'agent ;
- description du réseau d'accointances ;
- description ou instanciation des protocoles d'interactions.

Pour améliorer le processus de développement, une version de DIMA (appelée Méta-DIMA) fondée sur la métamodélisation est en cours de développement. L'objectif est d'offrir à un utilisateur, pas très familiarisé avec les systèmes multi-agents et les langages de programmation tels que JAVA, un outil pour décrire son système. Cette description est ensuite transformée en un système multi-agent. Ce travail est fondé sur l'outil MétaGen [SAH 98].

5.4.2.3. Applications

DIMA a été utilisée pour développer plusieurs applications réelles (ventilation artificielle [DOJ 97], simulation des modèles économiques [DUR 00], simulation d'un marché électrique, etc.) par les auteurs mais également par d'autres personnes n'ayant pas participé à son développement.

Ces applications peuvent être des simulations, des résolutions de problèmes ou des systèmes de contrôle ayant éventuellement des contraintes temps réel.

5.4.3. MASK (Multi-Agent System Kernel)

MASK est fondé sur l'approche VOYELLES [DEM 97] et constitue le support logiciel de cette méthode [OCC 00]. Il fournit des bibliothèques d'agents (A), de manipulation d'environnement (E), d'interaction (I) et d'organisation (O) ainsi que les outils d'aide à la programmation.

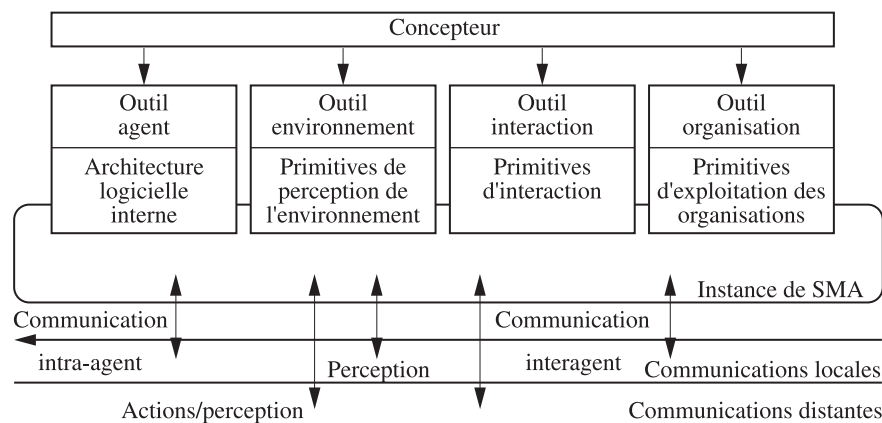


Figure 5.5. Vue générale de MASK

5.4.3.1. Agents

Plutôt que de proposer un modèle d'agent à dimension variable, MASK présente une approche multimodèle. Plusieurs modèles sont disponibles, les plus marquants étant ASTRO [OCC 98] (agent hybride) et PACORG [BAE 98] (agent réactif). Dans le même esprit, la structure des agents est complétée par des composants de base définis sur les boîtes à outil AEIO [OCC 01]. Ces différents composants s'intègrent aux architectures d'agents pour réaliser des structures d'agents opérationnelles. Les fonctions d'interaction sont fournies par les bibliothèques d'interaction. Pour le type de langage à protocole IL [DEM 95] par exemple, on disposera d'un ensemble de primitives d'exploitation de protocoles. On dispose aussi d'un modèle d'interaction par forces (PACO). Des primitives d'exploitation de la représentation externe de l'organisation selon le modèle RESO sont disponibles dans les bibliothèques d'organisation.

5.4.3.2. Outils de développement

Le principal but de MASK est de fournir au concepteur de nombreux outils intégrés au sein d'un même environnement (voir figure 5.5). MASK est en effet composé de boîtes à outils couvrant les différents aspects du paradigme multi-agent.

La boîte à outils *Agent* fournit à l'utilisateur des éditeurs d'agents instances de modèles d'agents prédéfinis ou permet au concepteur de modèles d'ajouter de nouveaux modèles avec leurs éditeurs associés.

La boîte à outils *Environnement* fournit à l'utilisateur des bibliothèques de fonctions pour manipuler les environnements (simulés) où évoluent les agents. Le concepteur peut définir de nouveaux environnements et associer les bibliothèques correspondantes. Cette boîte dépend très fortement des applications.

La boîte à outils *Interaction* est chargée de la présentation des fonctions d'interaction. Cette boîte se décompose en deux parties pour chaque modèle disponible : une bibliothèque de fonction et un éditeur d'interaction. Ainsi un éditeur de protocoles permettant de construire le protocole propre à notre application est fourni avec la boîte IL. On pourra aussi disposer par défaut des protocoles consacrés (Contract Net, entre autres).

La boîte à outils *Organisation* se décompose comme la boîte interaction en deux parties : un ensemble d'outils de modélisation de l'organisation fournissant des primitives d'exploitation et des éditeurs permettant d'instancier des organisations sur notre SMA. Le modèle RESO propose une interface graphique de construction de la représentation d'une organisation de SMA. De même, on pourra fournir par défaut les organisations classiques telles que la hiérarchie simple.

Des outils d'aide au choix des modèles sont envisagés. Les éditeurs particuliers permettent la construction des éléments des SMA (avec quelques automatisations partielles) et aide à la génération du SMA opérationnel. Un outil de validation des protocoles (AGIP) [KON 98] est disponible (en cours d'intégration).

On peut construire de façon indépendante les parties déclaratives de chacune des instances d'environnement, interaction et organisation. En fonction des types d'agent choisis, un outil d'assemblage adapté est lancé pour intégrer les AEIO. Dans le cadre des modèles d'agents cognitifs, peu nombreux et complexes, l'outil d'assemblage guide la construction des parties déclaratives de l'agent (spécification des plans, etc.) et assiste la programmation des parties opératoires par l'aide à l'intégration des primitives des autres bibliothèques choisies. Dans le cas des modèles d'agents réactifs, l'outil d'assemblage propose l'intégration automatique des AEIO. La programmation des agents réactifs se fait par configuration des paramètres qui gèrent leur décision. Après instanciation des EIO, l'outil propose une instanciation automatique des agents (leur nombre interdit habituellement dans ce contexte de les créer à la main) et une configuration automatique des paramètres.

5.4.3.3. Applications

MASK est un prototype de recherche. La plate-forme a été principalement utilisée pour simuler quelques applications dans le domaine des systèmes coopératifs (agent hybrides) ou pour des applications de résolution de problèmes en cartographie (agents réactifs). Sa structure très ouverte permet de viser de nombreux domaines d'application, sous réserve pour le concepteur de définir ses propres bibliothèques d'environnement.

5.4.4. Zeus

Zeus [NWA 99] est un environnement conçu et réalisé par British Telecom pour développer des applications collaboratives. Il est fondé sur les travaux de la FIPA.

5.4.4.1. Agents

L'architecture des agents ZEUS est similaire à la majorité des architectures d'agents collaboratifs. Elle regroupe principalement les composants suivants :

- une boîte aux lettres et un gestionnaire de messages qui analyse les messages de la boîte aux lettres et les transmet aux composants appropriés,
- un moteur de coordination,
- un *planner* et un *scheduler* qui planifie les tâches de l'agent en fonction des décisions du moteur de coordination, des ressources disponibles et spécifications des tâches,
- plusieurs bases de données représentant les accointances de l'agent, l'ontologie utilisée, etc.,
- un contrôleur d'exécution qui gère l'horloge locale de l'agent et les tâches actives.

5.4.4.2. Outils de développement

Zeus offre une méthodologie de développement principalement fondée sur la notion de rôle. Cette méthodologie comprend quatre étapes.

L'analyse du domaine consiste à comprendre et modéliser le domaine. Cette étape est principalement fondée sur la notion de rôle. Le modèle de rôle définit la manière dont sont définis et liés les rôles, sans se préoccuper du comportement effectif des agents.

La conception d'agents permet de dériver le comportement de l'agent en fonction du rôle. Cette étape permet, en effet, d'étudier la réutilisabilité des solutions existantes (ontologies, services...).

La réalisation détermine l'implémentation du modèle conceptuel.

Le suivi d'exécution permet à l'utilisateur de suivre l'exécution afin de tester, déboguer et optimiser.

Des éditeurs sont associés à chacune de ces étapes. Par exemple, dans la partie analyse, un éditeur de modèle de rôle est fourni. Cet éditeur est issu du diagramme de classes de UML [JAC 99].

5.4.4.3. *Applications*

Zeus a été utilisé pour développer plusieurs applications réelles (vente aux enchères, simulation de la fabrication des ordinateurs, etc.).

Les caractéristiques des domaines d'applications de Zeus ont été clairement définies par les concepteurs. Parmi ces caractéristiques :

- chaque agent crée un plan qui nécessite un raisonnement explicite pour atteindre son but ;
- la résolution de problème nécessite une coopération entre agents ;
- le rôle de chaque agent consiste à contrôler un système externe qui réalise une tâche du domaine d'application, la résolution de problème est ainsi effectuée par ce système externe et contrôlée par les agents.

5.4.5. *Discussion*

Le processus de développement d'un système multi-agent comprend plusieurs étapes dont la complexité est liée, d'une part, au domaine d'application et, d'autre part, à l'architecture d'agents. L'automatisation de chacune de ces étapes est un processus complexe. Les solutions proposées diffèrent en effet d'un environnement à un autre. Ceci est principalement dû au fait que ces environnements sont souvent une abstraction de travaux antérieurs. Par exemple, Jack est la continuité des travaux sur les architectures d'agents BDI et Cormas est une abstraction de plusieurs applications (des simulations). Les environnements résultats sont, en effet, plus appropriés au développement d'applications similaires et les systèmes multi-agents développés ont souvent des caractéristiques similaires.

D'autre part, la majorité des environnements de développement existants traitent la conception et/ou l'implémentation. La validation des systèmes multi-agents développés avec ces différents environnements est souvent expérimentale.

Plusieurs communautés (par exemple, celle des spécialistes de l'ingénierie des connaissances) ont évoqué l'approche multi-agent pour modéliser ces processus.

Cette solution a été également adoptée par L. Gasser dans MACE, mais aucun environnement ne l'a malheureusement reprise.

5.5. Vers des AGL multi-agents

Ces dernières années ont vu fleurir plusieurs environnements de développement. Cependant, dans la majorité de ces environnements, les entités appelées agents sont des objets mobiles. Ces objets mobiles n'ont pas les caractéristiques principales des agents (autonomie, proactivité, etc.).

Dans ce chapitre, nous avons donc choisi de ne citer que les environnements multi-agents. Ces derniers sont souvent fondés sur des propositions conceptuelles très intéressantes. Ces propositions sont issues de longs travaux sur les systèmes multi-agents.

Cet état de l'art montre que le processus de développement des systèmes multi-agents n'est pas encore complètement opérationnel. Mais il confirme également que l'on peut commencer à développer des applications multi-agents avec les premières versions de ces environnements dont la couverture des étapes du cycle de vie progresse peu à peu.

Un bon outil serait celui qui répondrait le mieux aux défis que nous avons présentés en introduction de ce chapitre :

- un bon environnement doit être ouvert. Il peut être basé sur des composants comme MALEVA [LHU 98] ou JAFIMA [KEB 95]. Sa structure peut même adopter des principes d'auto-organisation multi-agent comme dans MACE ;
- un bon environnement doit intégrer l'hétérogénéité. Il peut intégrer une vision multimodèle de la conception de SMA comme dans MASK. Il peut offrir un modèle multiparadigme à géométrie variable comme dans DIMA ;
- un bon environnement doit fournir des outils de suivi ou de simulation comme dans CORMAS ;
- un bon environnement doit permettre de valider partiellement. C'est le point qui mérite toute l'attention dans les recherches à venir ;
- un bon environnement doit offrir des capacités techniques de déploiement pour répondre aux besoins d'exécution distribuée et de sécurité. Dans ce domaine, et même si leur objectif est quelque peu différent, il est intéressant de faire référence aux systèmes de développement d'agent mobiles, comme Voyager [VOY 98], qui se sont particulièrement penchés sur ces problèmes.

C'est, l'a vu ci-dessus, de la mise en commun des aspects présentés par de nombreux outils existants que pourrait résulter l'atelier idéal de génie logiciel multi-agent.

5.6. Bibliographie

- [ABC 99] ABCHICHE N., Elaboration, implémentation et validation d'une approche distribuée pour l'intégration de modèles de raisonnement hétérogènes : application au diagnostic de pannes électriques, Thèse de doctorat, Université Paris 8, 1999.
- [AFI 99] BOISSIER O., GUESSOUM Z., OCCELLO M., « Plates-formes multi-agent », *Bulletin AFIA*, n° 39, octobre 1999.
- [AGE 99] AGENT BUILDER, Agent Builder User's guide, Documentation externe, <http://agentbuilder.com/>, 1999.
- [AGR 87] AGREE P., CHAPMAN D., « Pengi: An implementation of a theory of activity », dans *AAAI'87*, vol. 1, p. 268-272, Seattle, 1987.
- [BAE 98] BAEIS C., Fonctionnalité Emergente dans une Société d'Agents Autonomes, Thèse de doctorat, Institut national polytechnique de Grenoble, 1998.
- [BEN 97] BENSaid N., MATHIEU P., « A Framework for Cooperation in Hierarchical Multi-Agent Systems », dans *Mathematical Modelling and Scientific Computing*, vol. 8, 1997.
- [BOI 96] BOISSIER O., DEMAZEau Y., « ASIC: An architecture for social and individual control and its application to computer vision », dans J. Perram et J.P. Müller, *Distributed Software Agents and Applications, 6th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW'94*, Springer-Verlag, LNAI 1069, p. 135-149, mai 1996.
- [BOU 92] BOURON T., Structures de communication et d'organisation pour la coopération dans un univers multi-agent, Thèse de doctorat, Université Paris 6, Laforia, 1992.
- [BOU 98] BOUSQUET F., BAKAM I., PROTON H., LEPAGE C., « Cormas: Common-Pool Resources and Multi-Agent Systems », *Actes de la 11^e Conférence Internationale sur les applications industrielles et d'ingénierie de l'Intelligence Artificielle et des Systèmes Experts*, Benicàssim, Castellon, Espagne, Springer-Verlag, Berlin, LNAI 1416, p. 826-37, 1988
- [BRA 95] BRAZIER F., DUNIN-KEPLICZ B., JENNINGS N., TREUR J., « Formal specification of multi-agent systems: a real world-case », dans *Proc. of 1st Int. Conf. on Multi-Agent Systems - ICMAS'95*, p. 25-32, San Francisco, Californie, juin 1995.
- [BRI 00] BRIFFAULT X., GUICHARD N., COEFFIER R., « La plate-forme MERCURE », dans *Systèmes multi-agents, méthodologie, technologies et expériences*, Hermès, p. 273-276, 2000.
- [BRI 89] BRIOT J.P., « Actalk: a testbed for classifying and designing actor languages in the smalltalk-80 environment », dans *Proc. of ECOOP'89*, Springer-Verlag, LNAI n° 1069, p. 109-129, Nottingham, 1989.

- [BRO 86] BROOKS R.A., « A robust layered control system for mobile robot », *IEEE Journal of Robotics and Automation*, vol. 2, n° 1, avril 1986.
- [BUS 00] BUSETTA P., RÖNNQUIST R., HODGSON A., LUCAS A., JACK Intelligent Agents – Components for Intelligent Agents in Java, Technical report, 2000 (<http://www.agent-software.com.au/>).
- [CHE 93] CHEVRIER V., Etude et mise en œuvre du paradigme multi-agent : de ATOME à GTMAS, Thèse de doctorat, Université de Nancy I, 1993.
- [DAV 94] DAVIES W., EDWARDS P., « Agent-k: An integration of AOP and KQML », dans T. Finin et Y. Labrou (dir.), *Proceedings of the CIKM '94 Workshop on Intelligent Agents*, Gaithersburg, Maryland, décembre 1994.
- [DEC 90] DECKER K., GARVEY A., HUMPHREY M., « Effects of parallelism on blackboard system scheduling », dans *Proc. of the 4th Workshop on Blackboard Systems*, Boston, Massachusetts, 1990.
- [DEM 93] DEMAIZEAU Y., « La plate-forme PACO et ses applications », dans *Actes de la 2^e Journée du GDR-PRC IA*, Montpellier, décembre 1993.
- [DEM 95] DEMAIZEAU Y., « From Interactions to Collective Behaviour in Agent-Based Systems », *Proceedings of the 1st European Conference on Cognitive Science*, Saint-Malo, France, p. 117-132, avril 1995.
- [DEM 97] DEMAIZEAU Y., « Steps toward multi-agent oriented programming », *Proc. of the 1st International Workshop on Multi-Agent Systems, IWMAS '97*, Boston, Massachusetts, octobre 1997.
- [DIV 98] D'INVERNO M., KINNY D., LUCK M., WOOLRIDGE M., « A formal specification of dmars », dans Woolridge Singh, Rao (dir.), *Proceedings of ATAL98 Workshop on Agent Theories, Architectures, and Languages*, p. 155-176, Springer-Verlag, LNAI 1365, Amsterdam, Pays-Bas, août 1998.
- [DOJ 97] DOJAT M., PACHET F., GUESSOUM Z., TOUCHARD D., HARF A., BROCHARD L., « NéoGanesh: a Working System for the Automated Control of Assisted Ventilation in ICUs », *Artificial Intelligence in Medicine*, vol. 11, n° 2, septembre-octobre 1997.
- [DUR 88] DURFEE E.F., *Coordination of Distributed Problem Solvers*, Kluwer Int., Series in Engineering and Computer Science, 1988.
- [DUR 00] DURAND R., GUESSOUM Z., « Competence systemics and survival, simulation and emperical analysis », dans *Competence 2000 Conference*, Helsinki, Finlande, p. 10-14, juin 2000.
- [FER 91] FERBER J., CARLE P., « Actors and agents as reflective concurrents objects: a mering IV perspective », *IEEE Transaction on Systems, Man and Cybernetics*, vol. 21, n° 6, 1991.
- [FER 92] FERGUSON I.A., *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*, PhD thesis, University of Cambridge, Clare Hall, 1992.
- [FER 95] FERBER J., *Les systèmes multi-agents, vers une intelligence collective*, InterEditions, Paris, 1995.

- [FER 98] FERBER J., GUTKNECHT O., « Alaadin: a meta-model for the analysis and design of organizations in mutli-agent systems », dans Y. Demazeau (dir.), *ICMAS'98*, p. 128-135, Paris, 1998.
- [FIS 94] FISHER M., « Temporal Logic », *A Survey of Concurrent MetaTEM- The Language and its Applications*, p. 480-505, Springer-Verlag, Heidelberg, 1994.
- [GAS 87] GASSER L., BRAGANZA C., HERMAN N., « Distributed Artificial Intelligence », dans *MACE: A Flexible Testbed for Distributed AI*, Pitman, 1987.
- [GAU 97] GAUVIN D., « Lalo », dans *Actes des 5^e Journées Francophones IAD et SMA*, Hermès, avril 1997.
- [GIR 93] GIROUX S., Agents et systèmes, une nécessaire unité, PhD thesis, Université de Montréal, 1993.
- [GRA 00] GRAHAM J., DECKER K., « Towards Distributed, Environment Centered Agent Framework Appearing », dans Nicholas Jennings et Yves Lesperance (dir.), *Intelligent Agents IV, Agent Theories, Architectures, and Languages*, Springer-Verlag, 2000.
- [GUE 97] GUESSOUM Z., « DIMA : Une plate-forme multi-agent en Smalltalk », *Revue Objet*, vol. 3, n° 4, décembre 1998.
- [GUE 99] GUESSOUM Z., BRIOT J.-P., « From active objects to autonomous agents », *IEEE Concurrency*, vol. 7, n° 3, p. 68-76, 1999.
- [GUE 00] GUESSOUM Z., « A Multi-Agent Simulation Framework », *Transactions of Computer Simulation*, vol. 17, n° 1, p. 2-11, avril 2000.
- [HAN 99] HANNOUN M., BOISSIER O., SICHMAN J., SAYETTAT C., « MOÏSE : un modèle organisationnel pour la conception de systèmes multi-agents », dans *Actes des 7^e journées francophones IAD SMA*, La Réunion, 1999.
- [HAY 95] HAYES-ROTH B., « An architecture for adaptative intelligent systems », *Artificial Intelligence*, vol. 72, n° 1-2, p. 329-365, janvier 1995.
- [HUN 95] HUNTBACH M., JENNINGS N., RINGWOOD G.A., « How agents do it in stream logic programming », dans *Proc. of 1st Int. Conf. on Multi-Agent Systems – ICMAS'95*, San Francisco, Californie, juin 1995.
- [JAC 99] JACOBSON I., BOOCH G., RUMBAUGH J., *The Unified Software Development Process*, Addison-Wesley, 1999.
- [KEN 95] KENDALL E.A., MALKOUN M.T., JIANG C.H., « A methodology for developping agent based systems », dans C. Zhang et D. Lukose (dir.), *Distributed Artificial Intelligence: Architecture and Modelling, Proceedings of First Australian Workshop on DAI*, Canberra, Australie, Springer-Verlag, LNAI 1087, p. 85-99, novembre 1995.
- [KON 98] KONING J.L., FRANÇOIS G., DEMAZEAU Y., « An approach for designing negotiation protocols in a multiagent system », dans José Cuenca (dir.), *15th IFIP World Computer Congress, IT&KNOWS Conference*, Austrian Computer Society, Vienne (Autriche), Budapest (Hongarie), septembre 1998.

- [KUW 95] KUWABARA K., ISHIDA T., OSATO N., « Agentalk: Describing multiagent coordination protocols with inheritance », dans *Proc. 7th IEEE International Conference on Tools with Artificial Intelligence, ICTAI'95*, p. 460-465, 1995.
- [LAA 88] LAASRI H., MAITRE B., HATON J.-P., « Organisation, coopération et exploitation des connaissances dans les architectures de blackboard: Atome », dans *Actes des 8^e Journées Systèmes Experts et leurs Applications – Volume 3*, p. 371-390, Avignon, 1988.
- [LHU 98] LHUILLIER M., Une approche à base de composants logiciels pour la conception d'agents. Principes et mise en œuvre à travers la plate-forme MALEVA, Thèse de doctorat, Université de Paris VI, 1998.
- [LUC 95] LUCK M., D'INVERNO M., « A formal framework for agency and autonomy », dans *Proc. of 1st Int. Conf. on Multi-Agent Systems – ICMAS'95*, San Francisco, Californie, juin 1995.
- [MAR 89] MARUICHI T., Organization Computation, A Framework for Distributed Cooperative Problem Solving Using Autonomous Agents and their Groups, PhD thesis, Keio University, 1989.
- [MAR 97] MARCENAC P., « Modélisation de systèmes complexes par agents », *Technique et Science Informatiques*, vol. 16, n° 8, p.1013-1038, octobre 1997.
- [MIN 96] MINAR N., BURKHART R., LANGTON C., ASKENAZI M., *The swarm simulation system: a toolkit for building multi-agent simulations*, <http://www.santafe.edu/project/swarm>, 1996.
- [MUL 94] MÜLLER J., PISCHEL M., « Modelling reactive behaviour in vertically layered agent architectures », dans A. G. Cohen (dir.), *Eleventh European Conference on Artificial Intelligence (ECAI'94)*, p. 709-713, Amsterdam, Pays-Bas, 1994.
- [NWA 99] NWAMA H., NDUMU D., LEE L., COLLIS J., « Zeus: A tool-kit for building distributed multi-agent systems », *Applied Artificial Intelligence Journal*, vol. 13, n° 1, p.129-186, 1999.
- [OBJ 98] OBJECTSPACE, *Voyager Core Technology 2.0 User Guide*, ObjectSpace, 1998.
- [OCC 93] OCCELLO M., Blackboards distribués et parallèles : Application au contrôle de systèmes dynamiques en robotique et en informatique musicale, Thèse de doctorat, Université de Nice-Sophia Antipolis, janvier 1993.
- [OCC 98] OCCELLO M., DEMAZEAU Y., BAEIJS C., « Designing organized agents for cooperation in a real time context », dans *Collective Robotics*, Springer-Verlag, LNCS/LNAI 1456, p. 25-37, 1998.
- [OCC 00] OCCELLO M., KONING J.L., « Multiagent oriented software engineering an approach based on model and software reuse », dans *Proc. of 15th European Meeting on Cybernetics and Systems Research, 2nd International Symposium From Agent Theory to Agent Implementation*, p. 645-657, Vienne, Autriche, 2000.
- [OCC 01] OCCELLO M., KONING J.L., BAEIJS K.K., « Conception de systèmes multi-agents: quelques éléments de réflexion méthodologique », *Technique et Science Informatiques*, vol. 20, n° 2, p. 233-264, Hermès, Paris, 2001.

- [SAH 98] SAHRAOUI H.A., REVAULT N., BLAIN G., PERROT J.-F., « Un outil pour la conception de bases de données à objets », *Technique et Science Informatiques (TSI)*, vol. 17, n° 7, p. 839-868, 1998.
- [SHE 96] SHEN W., BARTHÈS J.-P., « An experimental multi-agent environment for engineering design », *International Journal of Cooperative Information Systems*, vol. 5, n° 2-3, p. 131-151, 1996.
- [SHO 93] SHOHAM Y., « Agent oriented programming », *Artificial Intelligence*, vol. 60, n° 1, p. 51-92, mars 1993.
- [THO 95] THOMAS S., « The placa agent programming language », dans M. Woolridge et Jennings N. (dir.), *Proc. of ECAI-94 ATAL Workshop on Agent Theories, Architectures, and Languages*, Amsterdam, Springer-Verlag, LNAI 890, p. 355-370, août 1995.
- [VIN 01] VINCENT R., HORLING B., LESSER V., *An Agent Infrastructure to Build and Evaluate Multi-Agent Systems: The Java Agent Framework and Multi-Agent System Simulator* Kluwer Academic Publishers, 2001.
- [WHI 98] WHITE T., PAGUREK B., « Towards multi-swarm problem solving network », dans Y. Demazeau (dir.), *ICMAS'98*, p. 333-340, Paris, 1998.
- [WOO 95] WOOLRIDGE M., JENNINGS N., « Intelligent agents: Theory and practice », *The Knowledge Engineering Review*, vol. 10, n° 2, p. 115-152, 1995.

Chapitre 6

Application des systèmes multi-agents dans les télécommunications

6.1. Introduction

6.1.1. *Evolutions des télécommunications*

Les télécommunications, au cours de la dernière décennie, ont été l'objet de plusieurs évolutions marquantes, parmi lesquelles la disparition progressive en Europe des monopoles, l'explosion des marchés des mobiles et de l'internet, le déploiement de réseaux hauts débits (ATM, ADSL) et de réseaux intelligents (RI). Ces évolutions ont eu plusieurs effets. Elles ont notamment introduit une conception décentralisée des services dans le contexte du *web*, créé des nouveaux services de médiations et engendré l'apparition de fournisseurs de services réseaux qui ne disposent pas de leurs propres ressources réseaux. Ces évolutions ont également ouvert de nouvelles perspectives pour la décennie à venir, telles que la création de marchés ouverts dans le domaine des services de télécommunication et du commerce électronique, l'industrialisation de mobiles de troisième génération avec des capacités haut débit, la convergence des services mobiles et internet. Les valeurs suivantes illustrent ces prévisions d'évolutions : le débit des mobiles va être démultiplié par 200 dans les cinq prochaines années. Il passera de 9,6 kbit/s en 1999 (GSM) à 100 kbit/s (GPRS) en 2002 et atteindra 2 Mbit/s en 2004. En France, le parc de mobiles a dépassé le taux de pénétration du téléphone fixe en octobre 2001. Le chiffre d'affaires du commerce électronique pourrait représenter plus de 10 % du

commerce mondial en 2010. Pour répondre à ces évolutions, les recherches dans le domaine des télécommunications sont de plus en plus guidées par les services. Elles s'orientent vers le développement de services à forte valeur ajoutée, vers la réduction des temps de développement de services, vers le développement de la convergence entre fixe, mobile, son et donnée. Au niveau logiciel, les recherches s'orientent vers le développement d'architectures logicielles intégrées interfacées au réseau (les *middlewares*). L'objectif de ces recherches est d'étendre les rôles traditionnels des opérateurs de télécommunication, de produire des services innovants et de concevoir des infrastructures permettant de développer rapidement des services en réponse aux offres concurrentes.

6.1.2. Intérêts des SMA pour les télécommunications

Derrière ces objectifs, deux enjeux nous semblent fondamentaux dans le domaine des télécommunications : maîtriser la sémantique de l'information communiquée et passer de logiciels fournissant des fonctionnalités à des logiciels offrant des services. Ces deux notions sont en partie considérées par les SMA et les agents logiciels, d'où, à notre avis, l'intérêt qui leur est accordé dans le domaine des télécommunications. Les techniques agent contribuent en effet, d'une part, au développement d'une ingénierie des connaissances permettant de ramener le traitement de l'information à un raisonnement sur des connaissances et, d'autre part, au développement de techniques de génie logiciel adaptées à la fourniture de services. Pour évaluer quelles pourraient être les implications des agents dans les télécommunications, il est nécessaire de préciser non seulement l'apport des SMA mais également de considérer la maturité des techniques agent. Le niveau de maturité d'une technologie peut être évalué en fonction des types d'expérimentations réalisées.

Niveau	Type d'expérimentation réalisée
n1	La conception de maquettes
n2	La réalisation de démonstrateurs fonctionnels
n3	La mise en œuvre de prototypes de laboratoire
n4	La conduite d'expérimentations sur sites
n5	Le déploiement industriel

Tableau 6.1. Niveaux de maturité d'une technologie

Les SMA étant particulièrement appropriés pour la restitution de comportements non déterministes et pour la modélisation de systèmes complexes dans des

environnements ouverts, il convient d'examiner les évolutions des télécommunications suivant ces perspectives. Le domaine des télécommunications offre des perspectives d'environnements ouverts, que ce soit au niveau du *web* ou des futurs services réseau. Il permet en outre d'explorer les différentes techniques d'agent : les agents mobiles, les assistants *web* et les agents raisonnant sur des connaissances. Le domaine des télécommunications devrait de ce fait permettre de développer des agents intégrant un raisonnement sur la connaissance dans la continuité de programmes fournissant un traitement « intelligent » de l'information.

6.1.3. Historique des recherches sur les SMA dans les télécommunications

Les premiers travaux de recherche sur l'application des SMA aux télécommunications ont été réalisés au début des années 1990. Les techniques agent se positionnaient alors comme un support possible pour l'interopérabilité de systèmes ouverts (KQML) [FIN 97] et comme une solution pour la résolution de problèmes distribués [BOU 92]. Les applications des SMA aux télécommunications réutilisent souvent des travaux antérieurs à 1990 notamment le protocole Contract Net [SMI 80] réalisé en 1980 dans le domaine de la reconnaissance de phénomènes à partir de senseurs répartis. Les premières applications des SMA dans les télécommunications ont porté sur la supervision distribuée de réseaux [PLU 96]. Elles ont été suivies en 1994 par des expérimentations de télématique utilisant la technologie d'agents mobiles Telescript [WHI 97]. Des logiciels d'agents mobiles ont été développés ultérieurement en Java [MER 98]. Le développement de techniques de génie logiciel applicables aux agents s'est inscrit dans la perspective d'un déploiement de ces premiers prototypes. En France, ils ont été dynamisés par un appel à projets publié par le centre de R&D de France Télécom sur le thème des agents en 1995. A partir de 1996, les télécommunications sont devenues un domaine d'application majeur des agents. Les services de recommandation sur le *web* assimilés à des agents se sont multipliés. Le processus de standardisation des technologies agents, FIPA [CHI 98], a été mis en place sur l'initiative d'industriels des télécommunications. Ce dynamisme a entraîné le lancement en 1998 de treize projets européens ACTS [MAG 99] sur l'application des agents au domaine des télécommunications. Le tableau 6.2 indique les grands domaines d'application couverts par les « agents » dans les télécommunications.

6.2. Applications des systèmes multi-agents dans le *web*

6.2.1. Services d'assistance dans l'organisation et le déroulement de voyages

Les premières expérimentations sur la réalisation de services d'assistance aux voyages fondés sur des agents ont été réalisées de 1994 à 1996 par les opérateurs

NTT, AT&T et France Télécom autour de la technologie Telescript. Les objectifs de ces expérimentations étaient, d'une part, d'intégrer différents moyens de notification tels que le fax, le *pager* et l'*e-mail* et, d'autre part, d'offrir des bouquets de services complémentaires prenant en compte le profil de l'utilisateur, tels que le calcul d'un itinéraire et la réservation de billets de transport. Ces services étaient les prémices de deux grandes classes de services d'intermédiation : les services de courtage et les services de personnalisation. Depuis 1996 plusieurs projets sur l'assistance aux voyages ont réalisé des démonstrateurs¹ [GER 01].

Domaines	Applications
Web et internet	Assistant pour l'organisation de voyages
	Médiation dans le commerce électronique
	Création de communautés <i>web</i>
Services et réseaux de télécommunication	Réseaux privés virtuels (VPN) et entreprises virtuelles
	Mobile de 3 ^e génération et réseau intelligent (RI)
	Supervision et gestion de réseaux
Ingénierie logicielle des télécommunications	Standardisation d'interopérabilité agent
	Méthodologie et atelier logiciel agent

Tableau 6.2. Domaines des télécommunications couverts par les projets « agents »

Scénario « assistance aux voyages »

M. Jean est un client du « service d'assistance aux voyages ». Il possède un ordinateur et un PDA (*Personal Digital Assistant*) intégrant un mobile. M. Jean utilise son ordinateur pour adresser des requêtes à son agent-logiciel « assistant personnel ». Ce logiciel connaît ses préférences et habitudes et peut ainsi personnaliser les requêtes. Il est de plus autorisé à prendre certains engagements, notamment à faire des opérations de réservation, en lieu et place de M. Jean. En son absence, il est capable de traiter les réponses aux requêtes. Le logiciel assistant personnel s'adresse à un ou plusieurs agent-logiciels de type courtier. Le logiciel courtier procède à une planification du voyage. Après avoir décomposé la requête en sous-requêtes en fonction de la segmentation du marché, il recherche plusieurs prestataires de services sur le *web* pouvant répondre à ces sous-requêtes. Si un logiciel courtier ne connaît pas de tels prestataires, il diffuse un appel à propositions sur le *web* ou consulte des annuaires. Les tarifs et caractéristiques du voyage sont ensuite négociés entre les logiciels courtier et fournisseur de services en fonction des

1. [Http://www.labs.bt.com/profsoc/facts](http://www.labs.bt.com/profsoc/facts).

priorités du client qui ont été communiquées par le logiciel assistant personnel. Au cours de son voyage, M. Jean utilise son PDA pour être informé d'événements non prévus qui peuvent perturber ou agrémenter son voyage et le cas échéant pour demander à son assistant personnel une nouvelle planification de la fin du voyage.

Ce scénario met en œuvre trois types d'agents : les assistants personnels, les courtiers et les prestataires de service. La fourniture du service est personnalisée. Elle est décomposée en deux phases. La première phase est l'organisation du voyage. Elle inclut la détermination de l'itinéraire, l'achat des billets de transport, la réservation des hôtels et le recueil d'information sur les divertissements. La deuxième phase couvre le suivi du bon déroulement du voyage et, si nécessaire, la réorganisation de certaines parties du voyage en fonction d'événements non prévus par exemple des difficultés dans les transports.

Deux exemples : les projets ACTS FACTS et MOTIV-PTA

Le projet ACTS FACTS² a réalisé un démonstrateur du service d'assistance aux voyages dans le but de valider les spécifications produites par le standard FIPA. Conformément au scénario précédemment décrit, ce démonstrateur considère trois types d'agents : les assistants personnels (PTA), les courtiers (TBA) et les fournisseurs de services (TSA). Ces derniers sont basés sur des services existants sur le *web* (compagnies aériennes, chaînes hôtelières). Pour mieux tester l'interopérabilité, les trois types d'agents ont été développés par différents partenaires, respectivement : Broadcom, British Telecom et France Télécom. De plus, l'exécution des agents est supportée par deux plates-formes interopérables, Jade [BEL 99] et ASL. Des assistants portables utilisant le protocole WAP sont utilisés pour accéder au service au cours du voyage.

L'interopérabilité des agents ACTS FACTS est assurée par l'inclusion dans chaque type d'agent d'analyseurs (ACL et SL) et de générateurs de messages conformes à FIPA. En dehors de ces composants, l'architecture des agents varie en fonction du type d'agent considéré. Les agents de type PTA incluent : une *base de cas* où sont mémorisées des informations sur les précédentes requêtes d'un utilisateur, une *base de profils* où sont enregistrées les préférences déclarées par les utilisateurs ou apprises automatiquement, un *générateur de requêtes* qui crée et adresse des requêtes aux TBA à partir des demandes de l'utilisateur, des *connaissances* sur ses préférences et d'*informations* enregistrées dans la base de cas, un *analyseur d'offres* vérifiant l'adéquation des offres reçues en réponse aux requêtes. Les agents de type TBA sont des agents de type BDI. Ils sont composés d'une base de connaissance, d'un moteur de raisonnement incluant un planificateur et un *scheduleur*, d'un

2. Le projet ACTS FACTS (1998-2000) a été réalisé par British Telecom lab, Broadcom, France Télécom R&D, l'ONERA, le CSELT, Imperial College, Nortel, Alcatel, GMD.

analyseur de contenus, d'un gestionnaire de profils et d'une bibliothèque de protocoles FIPA. Le planificateur procède par propagation de contraintes. Les agents TSA incluent : une base de connaissances, un générateur de requêtes *web* et des gestionnaires de conversation, de notification, de réservation, de paiement.

Le projet Motiv PTA³ expérimente le précédent scénario sur la plate-forme agent MECCA développée par Siemens. Un prototype a été expérimenté sur trois sites (en Bavière, à Badden-Württemberg et à Hanovre-Brunswick). L'accès au système se fait *via* des postes fixes et des mobiles. Les réseaux de communication utilisés sont l'internet, GSM et ISDN. Ce prototype permet de planifier l'itinéraire en prenant en compte des informations sur : plusieurs modes de transports (la voiture, le train et l'avion), l'occupation des parcs de stationnement, les difficultés de circulation et les préférences de l'utilisateur. Le système est informé de la localisation du voyageur *via* le GSM. Il peut ainsi faire des déductions sur les retards et adapter en conséquence l'itinéraire initial. Certaines fonctionnalités du projet ont fait l'objet d'expérimentations avec une cinquantaine d'usagers.

Intérêts, techniques expérimentées, problèmes

Le scénario d'assistance au voyage présente plusieurs intérêts. Il illustre la relation établie par une conception agent entre la modélisation de services et le *business model* (les agents étant associés à des acteurs du marché bien identifiés). Il décrit un service susceptible d'être commercialisé et fait appel à des sources d'information et des techniques logicielles en grande partie disponibles.

Plusieurs prototypes ont mis en œuvre la phase 1 du scénario. Ils utilisent des protocoles de communication conformes à FIPA, des techniques de planification et de résolution de contraintes et des architectures *middleware* réparties. La maturité de ces technologies est de niveau (n3) (voir tableau 6.1). Les expérimentations sur site en sont encore à leurs balbutiements. L'utilisation de techniques issues de FIPA a permis d'aborder les problèmes liés à l'interopérabilité d'agents. Une partie de la phase 2 du scénario (la prise en compte de la localisation géographique du client dans l'accomplissement de services) a été testée dans le projet MOTIV-PTA.

Certains problèmes techniques n'ont pas été expérimentés, notamment la re-planification en temps réel et l'autorité des agents. Les démonstrateurs n'ont en outre pas testé les spécifications d'interopérabilité FIPA dans un environnement réellement ouvert, c'est-à-dire dans un environnement où les concepteurs d'agents ne peuvent pas se concerter. Les actes de langage ont été mis en œuvre uniquement

3. Le projet allemand MOTIV-PTA (1996-1999) a été réalisé dans le cadre du programme « *mobility in Urban Area* » par Siemens, Opel, BMW, Daimler-Benz, IBM, Bosch, Volkswagen, Debis System Haus, VDO Car Communication.

dans le cadre de protocoles. Le fait que chaque démonstrateur introduise sa propre ontologie du domaine d'application est une limite fondamentale à l'adaptation des agents dans un environnement ouvert. Dans les démonstrateurs actuellement réalisés, la définition des agents gagnerait à être accentuée par un traitement centralisé de l'information au sein de chaque agent, en vue de faire des recoupements d'information et de pouvoir automatiser les raisonnements élaborés à partir de connaissances extraites de ces informations.

6.2.2. Extensions de ces expérimentations au commerce électronique

L'utilisation des agents logiciels peut être généralisée à d'autres formes de vente, à l'achat de produits variés et s'inscrire par conséquent dans le cadre plus général du commerce électronique. Le *web* offre une zone commerciale d'étendue mondiale. Dans un environnement aussi étendu, un enjeu est d'optimiser la recherche de produits, la comparaison des offres et la négociation des termes d'achat. L'introduction de techniques agent et de SMA permettrait d'automatiser ces tâches. Les logiciels agent peuvent, par exemple, être utilisés pour trouver un livre ou un CD au meilleur prix, pour considérer simultanément l'achat d'une voiture et la souscription d'une assurance, ou encore pour automatiser une vente aux enchères sur le *web*. Plusieurs composants de services qui pourraient être intégrés à un SMA dans le cadre du commerce électronique ont déjà été réalisés [BOU 97, GUT 98]. Ils portent sur l'évaluation des offres de différents vendeurs, sur la négociation d'un produit et sur la mise en œuvre de places de marchés pour agents logiciels.

Scénario « place de marché virtuel »

M. Jean souhaite acheter un ordinateur et vendre sa collection de livres. Il dispose d'un agent-logiciel « assistant d'achat » dont le rôle est de rechercher des produits et de négocier leur achat. Cet assistant est capable d'adapter son comportement en fonction des caractéristiques des produits recherchés et des modes de vente proposés par les marchands. M. Jean adresse une requête à son agent assistant d'achat pour l'achat d'un PC. Celui-ci fait appel à un ou plusieurs agents logiciel « courtier » afin de comparer les offres de vente de différents marchands. Pour la vente de ses livres, M. Jean crée un agent-logiciel de type « bouquiniste » qui implémente des comportements et protocoles de vente adaptés aux agents logiciel déjà sur le marché. La programmation de cet agent est fondée sur des serveurs d'ontologie qui définissent formellement « un bouquiniste » à partir de concepts plus élémentaires tels qu'un livre, un produit, un prix. L'agent « bouquiniste » fait appel à un ou plusieurs agents logiciel « annonceur » chargés de publier les offres de vente. Les prix sont ajustés par l'agent en fonction de la demande. Pour certains livres, l'agent « bouquiniste » fait appel à un agent logiciel de « vente aux enchères ».

Places de marché

Les projets KASBAH et MARKET MAKER réalisés au MIT ont étudié les interactions entre les agents dans les processus d'adaptation de l'offre à la demande [GUT 98]. Ils ont réalisé à cet effet une place de marché qui automatise les processus d'achat et de vente entre des agents représentant des utilisateurs. Les agents essaient de vendre ou d'acheter par eux-mêmes des articles. Les prix sont ajustés en fonction de la demande. Les utilisateurs interviennent uniquement pour entériner les transactions. La plate-forme KASBAH a été connectée à des terminaux mobiles pour permettre à des utilisateurs faisant leurs courses dans une boutique réelle de comparer les prix aux offres proposées sur le *web*. Le projet ARCADIA⁴ [CAM 98] simule une place de marché pour l'achat et la vente de voitures par petites annonces. Le système complète les informations trouvées dans un forum de petites annonces par des informations complémentaires issues de services *web*. Les agents jouent un rôle de courtier. Ils intègrent et unifient les interfaces issues des services *web*. Ils assistent les acheteurs et vendeurs pour déterminer le prix du marché et notifier le dépôt d'annonces correspondant à des demandes.

Services de comparaison de vendeurs

Les services de comparaison de prix en temps réel sur le *web* tels que JANGO [BOU 97] se multiplient depuis 1996 suite à l'expérimentation BARGAINFINDER [BOU 97]. SHOPBOT [DOO 97] est un démonstrateur qui a la particularité d'apprendre, de façon autonome, à consulter de nouveaux sites et à acheter sur ces sites. Le démonstrateur a été testé sur une douzaine de services de vente en ligne. Dans la phase d'apprentissage, SHOPBOT teste les différents formulaires de saisie proposés par les services *web* sur des produits courants dont il connaît les propriétés (ordre de prix, nom du distributeur, etc.). Des heuristiques sont utilisées en complément pour filtrer les formulaires. Le projet TETE-A-TETE [GUT 98] réalisé au MIT propose une stratégie de comparaison d'offres portant sur le prix et la qualité de service. Ce projet est basé sur des techniques d'utilité multi-attribuées et de satisfaction de contraintes distribuées.

Services de vente aux enchères

En 1999, il existait près d'une centaine de services de vente aux enchères sur le *web* parmi lesquels E-BAY et AUCTIONBOT. Le service E-BAY vendait 1 million d'articles par mois et comptait 500 000 membres en 1998. AUCTIONBOT est un démonstrateur de vente aux enchères offrant une API permettant aux utilisateurs de créer leurs propres logiciels d'achat ou de vente capables de participer à des enchères.

4. Le projet ARCADIA (1995-1998) a été réalisé dans le cadre d'un contrat de recherche de France Télécom avec l'ONERA/CERT et l'université de Toulouse.

Publication d'ontologies

Une ontologie est définie par la spécification, parfois formelle, de la signification des termes d'une taxonomie donnée. Les termes sont définis de préférence à partir de termes fondamentaux communs à plusieurs domaines, plutôt que par l'introduction de termes spécifiques. Des recherches ont été conduites sur les ontologies depuis une quinzaine d'années en intelligence artificielle. Dans le cadre du commerce électronique, les ontologies permettent de décrire les produits et services dans des termes et concepts partagés par les différents acteurs du *web*. La définition d'un service en référence à une ontologie publique peut faciliter la promotion d'un service, son accès par des logiciels de courtage, de vente et d'achat. Plusieurs bibliothèques d'ontologies sont déjà accessibles en ligne (CYC, ONTOLINGUA [FRID 97]). Elles incluent des concepts se rapportant au commerce. Le concept de prix est par exemple défini dans l'ontologie « Financial » de CYC et dans l'ontologie « Product » de ONTOLINGUA. L'IEEE⁵ s'intéresse à la standardisation des concepts de base communs à un grand nombre d'ontologies.

Intérêts, techniques expérimentées et problèmes

L'intérêt des SMA dans ce contexte est très étroitement lié à l'évolution du commerce électronique vers un marché ouvert très libéral. Le scénario de « place de marché virtuel » s'inscrit d'ailleurs dans cette perspective. La question est de déterminer si ce qui c'est passé sur le *web* pour l'information peut se reproduire pour le commerce.

Plusieurs aspects du scénario de « place de marché virtuel » ont été réalisés. Leur maturité technologique est assez disparate (niveau n2 à n4 dans le tableau 6.1). Les fonctionnalités générales d'une telle place, notamment des comportements d'achats et de ventes automatisés, ont été introduites dans des prototypes tels que KASBAH. Des protocoles de communication adaptés à l'échange d'informations et à la négociation entre logiciels ont été spécifiés dans FIPA. Des serveurs d'ontologies ainsi que des sites de comparaison d'offres commerciales interfacés à des services *web* sont d'ores et déjà accessibles en ligne. SHOPBOT a montré que l'extraction de l'information contenue dans des sites peut être automatisée et que, par conséquent, il est possible de concevoir des services de comparaison d'offres qui soient adaptés à un environnement ouvert.

Les différents éléments de solution n'ont pas été intégrés les uns aux autres. Ainsi, les services de comparaison d'offres ne sont pas interfacés à des assistants d'achat. Les agents actuellement conçus ne font pas référence à des ontologies publiques. Celles-ci sont d'ailleurs encore en nombre restreint. Il n'est d'ailleurs pas

5. [Http://ltsc.ieee.org/suo](http://ltsc.ieee.org/suo).

certain que la définition d'ontologies, de façon formelle, permette de capturer toute la sémantique de ses concepts, d'où un risque résiduel de problèmes d'interopérabilité entre des programmes faisant référence à une même ontologie lorsqu'ils sont conçus de manière non concertée. Il convient par conséquent d'examiner comment deux agents peuvent ajuster automatiquement leurs ontologies respectives. Il est en outre possible que des logiciels utilisables uniquement dans le cadre du commerce électronique supplantent des SMA plus génériques.

6.2.3. *Application des agents à la création de communautés web*

Des communautés virtuelles se sont créées sur le *web* pour faciliter l'échange d'informations entre des cybernautes ayant des centres d'intérêts communs. Plusieurs services à base d'agents tels que FIREFLY ont été réalisés autour de cette idée depuis 1996 [BOU 97]. Ces services sont réalisés sur des techniques de filtrage. Ils ont développé une approche en marge des approches théoriques des SMA. Cette approche est efficace mais n'est pas ouverte. En complément de ces travaux, les projets FRIENDS [VAE 99] et RICOCHET [BOT 99] ont étudié l'apport de techniques SMA dans un tel contexte. Ils considèrent la création de communautés d'agents et la migration des agents entre les communautés.

Scénario « organisation dynamique de communautés virtuelles »

M. Jean a un « assistant de navigation » qui apprend ses centres d'intérêts en analysant le contenu textuel des sites *web* qu'il consulte. Cet apprentissage est renforcé par la prise en compte de ses appréciations. Le profil de M. Jean ainsi constitué est régulièrement comparé aux profils des autres utilisateurs pour créer dynamiquement des communautés d'intérêts communs. Chaque profil est représenté par un agent. Celui-ci est défini par un comportement lui permettant de se joindre à des groupes et si nécessaire de constituer de nouvelles communautés. Les profils sont également utilisés pour déterminer les services qui *a priori* devraient intéresser les utilisateurs et pour faire des opérations de *marketing* ciblées.

Projets FRIEND et RICOCHET

Le projet FRIENDS⁶ a étudié la dynamique de l'organisation de communautés virtuelles sur le *web* et le maintien de propriétés organisationnelles. Dans un contexte ouvert tel que le *web*, les processus d'organisation sont nécessairement décentralisés. Une solution consiste à les intégrer aux comportements des agents. Dans FRIENDS, le principe de base retenu pour décrire les comportements des agents

6. Le projet FRIENDS (1996-1999) a été réalisé dans le cadre d'un contrat de recherche de France Télécom avec le Leibniz (IMAG).

est « qui se ressemble s'assemble ». Une organisation est formalisée par une arborescence binaire. Elle est construite à partir d'un calcul de l'entropie. Des algorithmes globaux ont été utilisés parallèlement aux algorithmes distribués qui sont intégrés aux agents, pour tester le bien-fondé de ce type de représentation. Chaque utilisateur saisit ses centres d'intérêts et peut visualiser l'organisation. Si la communauté à laquelle il appartient ne lui convient pas, il peut modifier ses intérêts pour que l'agent le représentant se déplace et aille dans d'autres communautés d'intérêts. FRIENDS a été testé sur une cinquantaine d'utilisateurs. Il a également été utilisé pour identifier des thèmes (religion, linguistique, alimentation, télécommunication, etc.) à partir des descriptifs de cinq mille services *web*.

RICOCHET⁷ est un système de création de communautés d'utilisateurs basé sur des techniques SMA. Les profils des agents sont constitués par des informations sur les centres d'intérêts des utilisateurs, des données extraites des logs (temps et fréquence de connexion) et des informations personnelles (âge, adresse, etc.). Le calcul de proximité des agents est basé sur l'utilisation de thesaurus permettant d'évaluer la distance entre les mots-clés décrivant les centres d'intérêts. Le système est destiné à être mis en œuvre sur des fournisseurs d'accès à l'internet qui peuvent être répartis sur plusieurs sites. La constitution des communautés doit donc être décentralisée. Trois types d'agents sont considérés. L'agent « manager » maintient une liste des communautés existantes. Les agents « profil » représentent les utilisateurs et s'enregistrent dans des communautés. Les agents « group » gèrent les communautés. Si un agent n'arrive pas à s'inscrire dans un ou plusieurs groupes existants, il crée un nouveau groupe. La communication entre les agents est basée sur les actes de langage FIPA « query » et « inform ». Des expérimentations ont été conduites sur trois cents profils d'agents pour évaluer la stabilité et la convergence du système lors de la modification des profils des agents. La pertinence des groupes est évaluée à partir de la mesure de leur inertie.

Intérêts, techniques expérimentées et problèmes

L'introduction des SMA dans ce contexte permet de définir les individus non seulement par un jeu de données, caractéristique de leur profil, mais également par un comportement. Dans FRIENDS et RICOCHET, les agents ont un comportement leur permettant de se rapprocher de façon autonome des autres individus partageant des intérêts communs. L'introduction de SMA couplés à des services tels que FREFLY est également un moyen d'aborder l'interopérabilité de plusieurs services de recommandation.

D'une façon générale, les expérimentations sur les services de personnalisation ont permis d'éprouver des techniques de représentation de profils basées sur des

7. Le projet RICOCHET (1996-1999) a été réalisé à France Télécom.

listes de mots-clés pondérés et sur des techniques d'apprentissage automatique de préférences. Ces techniques ont été déployées. Leur maturité correspond au niveau n5 du tableau 6.1. La maturité des techniques développées dans FRIENDS et RICOCHET se situe quant à elle au niveau n2.

Des problèmes restent à résoudre. Les techniques de calcul d'entropie utilisées pour segmenter la population sont souvent mises en œuvre de façon centralisée et nécessitent de recalculer régulièrement intégralement la segmentation. Elles ne sont de ce fait que partiellement adaptées à un environnement ouvert.

6.3. SMA dans les services et réseaux de télécommunication

6.3.1. *Application des SMA aux réseaux privés virtuels*

Il y a plusieurs décennies, lorsque des sociétés souhaitaient disposer d'un réseau de télécommunication privé, elles utilisaient une infrastructure de télécommunication qui leur était propre. Par la suite, ces demandes ont été satisfaites par des réseaux privés constitués de liaisons louées à un opérateur. Ces liaisons qui sont en dehors des réseaux publics garantissent une qualité de service et une complète confidentialité des données échangées. Les réseaux privés virtuels (VPN) sont des offres de réseaux privés mises en œuvre sur les réseaux publics. L'offre VPN permet de répondre à une demande croissante de connexions temporaires, la bande passante inutilisée par une entreprise à un instant donné étant potentiellement disponible pour une autre utilisation. Les SMA peuvent être utilisés pour automatiser la négociation des ressources réseau dans ce contexte. Le projet EURESCOM P712 a exploré les aspects fonctionnels d'un tel service. Le projet ACTS FACTS a examiné l'utilisation de FIPA pour supporter l'interopérabilité de différents fournisseurs de ressources et de services dans ce contexte. Les VPN pourraient s'inscrire à terme dans le contexte plus général de la création d'entreprises virtuelles. Cette perspective a été considérée par le projet ACTS MIAMI⁸.

Scénario « marché d'allocation de VPN »

Le scénario est centré sur l'allocation dynamique de VPN permettant la collaboration d'utilisateurs au moyen d'applications multimédia et sur la négociation des bandes passantes des VPN. Les utilisateurs potentiels de ce service sont des professionnels nomades et des entreprises. Trois types d'agents sont considérés : les assistants personnels de télécommunication (PCA), les fournisseurs de services réseau (SPA) et les fournisseurs de ressources réseau (NPA). Les PCA assistent les utilisateurs dans la recherche et la sélection des offres de télécommunication des

8. <http://www.fokus.gmd.de/research/cc/ima/miami>.

SPA. Ils négocient la qualité du service et son coût. Les SPA déterminent les services réseaux requis. Ils identifient les niveaux de sécurité exigés et sélectionnent les offres des NPA. Les NPA proposent des offres de connexions réseau et s'engagent à établir ces connexions si leurs offres sont retenues. Ils sont interfacés avec le système de gestion du réseau de leur domaine. Les termes et connexions externes à un domaine sont négociés entre les NPA.

M. Jean demande à son PCA d'établir depuis un hôtel une réunion multimédia à partir de demain pour une durée d'environ 2 jours et pour un budget de 5 000 Euros. L'agent PCA demande à un annuaire les adresses des SPA. Il contacte ces SPA pour connaître leurs offres. Chaque SPA consulté fait une demande d'offre aux NPA dont il a connaissance. Il sélectionne ensuite les propositions des NPA et fait des réservations provisoires. Chaque SPA consulté formule en retour une proposition au PCA de M. Jean. L'agent PCA sélectionne alors l'offre la plus intéressante. Il peut négocier, si nécessaire, les termes de la fourniture des services réseaux avec les SPA. La négociation porte sur la restriction de la plage de mise à disposition d'un débit donné. Une fois la liaison établie, les SPA peuvent proposer une renégociation de ces termes.

Projets ACTS FACTS ET EURESCOM P712

Le projet ACTS FACTS a développé un démonstrateur autour de ce scénario en vue de tester les spécifications de FIPA. Des VPN sont alloués pour supporter un service de visioconférence. Chaque partenaire (Nortel, GMD-FOKUS et ALCATEL) développe ses propres agents PCA, SPA et NPA. L'exécution des agents est supportée par deux plates-formes respectivement développées par Nortel et IKV+ (Grasshopper). Les PCA déterminent de façon coopérative la date de la réunion en fonction d'informations extraites des agendas Outlook des utilisateurs. Le protocole FIPA-ContractNet est utilisé à cet effet. Une fois qu'un accord a été passé sur une date, un des PCA contacte un SPA pour établir la connexion en temps voulu. Si le SPA accepte le service, la connexion est établie automatiquement à la date retenue. Le SPA contacte les NPA pour obtenir l'adresse IP nécessaire à l'établissement de la connexion. La visioconférence est établie sur l'initiative du PCA par le logiciel Netmeeting.

Le projet européen EURESCOM P712⁹ a également développé un démonstrateur qui teste les fonctionnalités de ce scénario [GAR 98]. Le démonstrateur est basé sur les réseaux Internet et ISDN. Une architecture CORBA est utilisée pour réserver les ressources de l'ISDN. Les fournisseurs de services sont sur l'internet.

9. Le projet EURESCOM P712 (1997-1999) a été réalisé par : British Telecom, Deutsche Telekom, l'université technologique de Berlin, le CSELT, KPN, Telefonica et Broadcom.

Intérêts, techniques expérimentées et problèmes résidant

Le précédent scénario se situe dans un contexte très libéral des télécommunications. L'intérêt de l'automatisation de la négociation entre les différents acteurs est étroitement lié au nombre d'acteurs présents sur le marché et à la dissociation des opérateurs réseau et des fournisseurs de services de télécommunication.

La maturité des techniques SMA sur ce domaine d'application se situe au niveau de la mise en place de démonstrateurs fonctionnels (niveau n2 dans le tableau 6.1).

Pour réaliser ce type d'application, les agents doivent être capables de communiquer à propos de la fourniture de services de télécommunication. Or, actuellement, il n'existe pas de définition standard des termes d'un tel dialogue.

Entreprises virtuelles, ACTS MIAMI

L'intérêt d'une architecture SMA dans ce scénario pourrait être renforcé par l'intégration de VPN dans le contexte plus général de la création d'entreprises virtuelles. Une entreprise virtuelle est constituée par une fédération provisoire de compagnies et de membres indépendants. Il peut par exemple s'agir d'une université virtuelle proposant des séries de cours sur le *web*. La mise en place d'une telle entreprise est par définition basée sur une infrastructure réseau. Les VPN répondent de façon intrinsèque aux nécessités de virtualité et de confidentialité requises par ce réseau. Un des intérêts de développer le concept d'entreprise virtuelle pour les industriels des télécommunications est donc de stimuler le marché des VPN. Les agents peuvent être utilisés pour l'approvisionnement et l'administration des VPN et pour supporter l'établissement d'accords entre les partenaires lors de la constitution de l'entreprise virtuelle.

Le projet MIAMI¹⁰ a spécifié un scénario envisageant la création d'une entreprise virtuelle dans le cadre de la mise en place de l'Euro, pour répondre aux demandes d'information à travers la diffusion de cours multimédia interactifs. L'infrastructure d'agents mobiles développée dans MIAMI offre un support pour l'identification de partenaires, ainsi que pour les négociations et l'administration de l'entreprise virtuelle. Elle est basée sur la plate-forme GRASSHOPPER et apporte des extensions au standard d'agents mobiles MASIF [MAG 98] au niveau du transport, de la communication et de la sécurité.

10. Le projet ACTS MIAMI (1998-2000) a été réalisé par : Alcatel, GMD-Fokus, France Télécom R&D, University College London, Algosystem CA, Hitachi, CRIN.

6.3.2. Utilisation de SMA dans le cadre des mobiles de troisième génération et du RI

Les réseaux mobiles de troisième génération vont être déployés de façon progressive à partir de 2003. Leurs débits pourront atteindre 2 Mbits/s. Ils seront de ce fait adaptés à l'utilisation d'applications multimédia et permettront d'accéder à l'ensemble des services de l'internet. Les caractéristiques des mobiles de troisième génération sont définies, en Europe, par la norme UMTS (*Universal Mobile Telecommunication System*). Cette norme introduit notamment les infrastructures VHE (*Virtual Home Environment*) et PCS (*Personnal Communication Support*). L'infrastructure VHE peut être vue comme un *middleware* qui masque à l'utilisateur d'un mobile lors du changement de domaine (*roaming*) les capacités réelles du réseau. Elle permet à un utilisateur d'accéder à l'ensemble des services de télécommunication fournis par son abonnement (ou à défaut à des services cohérents avec ceux-ci) même lorsqu'il est en dehors de la zone de couverture du domaine de son opérateur. L'infrastructure PCS définit des polices personnalisées pour traiter les appels en fonction de critères variés tels que l'heure de réception, l'émetteur et le contenu. Les innovations de service rendues possibles par l'évolution des réseaux nécessitent en retour une évolution des réseaux au niveau de l'administration et du contrôle de services. Des évolutions ont d'ores et déjà été apportées dans le cadre des réseaux intelligents. De nouvelles évolutions doivent être considérées pour répondre à la convergence des réseaux mobiles et de l'internet et à l'usage international de mobiles. Les SMA, souvent couplés à des fonctionnalités d'agents mobiles, apportent des éléments de solution à ces évolutions [GAL 00].

Scénario type « VHE et PCS dans l'UMTS »

M. Jean est un abonné de l'opérateur A-Telecom (son *network home*). Grâce au VHE, il peut, où qu'il se trouve même sur le domaine de couverture de l'opérateur C-Telecom, appeler automatiquement des numéros sélectionnés dans son carnet d'adresses ou encore, alors que A-Télécom utilise le « 10 » pour appeler les urgences et C-Telecom le « 115 », appeler le service local des urgences en tapant quel que soit sa localisation l'indicatif « 10 ».

Les opérateurs A-Telecom et C-Telecom mettent en œuvre des réseaux mobiles de troisième génération avec des infrastructures VHE et PCS. Leur architectures réseau sont hétérogènes. Seuls les services les plus récents sont mis en œuvre sur un réseau intelligent. L'infrastructure du VHE et du PCS de A-Telecom et C-Telecom est basée sur un *middleware* agent. Des agents associés aux utilisateurs adaptent la fourniture de services au fur et à mesure de leurs déplacements en fonction des changements de réseaux et de domaines.

Les agents déterminent comment exécuter les services des utilisateurs en fonction de leurs profils, des services auxquels ils sont abonnés, des services disponibles sur le domaine et de l'infrastructure réseau où ils se situent. Des agents

intègrent par ailleurs les polices personnalisées de gestion des communications entrantes. Une extension du scénario consiste à introduire la métaphore « d'agents mobiles » pour appréhender plusieurs fonctionnalités caractéristiques du VHE :

- lors du passage d'un domaine à l'autre une grande partie des services utilisant les données sur les profils utilisateurs reste accessible ;
- les temps de réponse ne sont pas altérés par la mobilité de l'utilisateur comme si le traitement des données sur les profils était réparti en fonction de la localisation du client ;
- lorsque des composants logiciels interfacés à d'autres composants logiciels ne sont pas disponibles sur un domaine de réseau, ces composants sont transférés d'un domaine à l'autre.

Projets EURESCOMP712, EURESCOMP810

Actuellement il n'existe pas d'expérimentations achevées du scénario « VHE et PCS dans l'UMTS ». Les principales fonctionnalités ont été spécifiées dans le projet EURESCOM P810¹¹ [GAL 00]. Certains problèmes ont été déclinés pour être étudiés dans le projet EURESCOM P712.

Le projet EURESCOM P810 a examiné l'applicabilité des nouvelles techniques dans le contexte de l'UMTS [UMT 95] et notamment l'utilisation de techniques agent dans une plate-forme de contrôle de services. L'intérêt des techniques agent est justifié à partir de l'identification des propriétés requises par l'infrastructure de services pour supporter les services VHE et PCS. Le projet propose de remplacer une partie des structures globales tels que les registres *Home Location Register* et *Visited Location Register* qui sont utilisées dans les mobiles de deuxième génération pour le contrôle de services, par une architecture multi-agent supportant les services PCS et VHE. Les agents sont chargés de construire dynamiquement les exécutables des services. Les composants de services sont par ordre de priorité : invoqués sur le domaine où se situe l'utilisateur, téléchargés depuis les autres domaines, accessibles *via* des appels procéduraux distants. Le projet a également considéré l'intégration de l'exécution des agents dans l'architecture réseau.

Le projet EURESCOM P712 considère les techniques agent pour faciliter, dans le domaine des télécommunications, l'intégration de nouveaux composants logiciels aux composants existants [CAR 98]. Le problème est décliné dans le contexte de la messagerie électronique. Une architecture multi-agent est définie pour automatiser la recherche d'un décodeur approprié, assurer son installation automatique, convertir le

11. Le lot d'EURESCOM P810 (1998-2000) concerné a été réalisé par : GMD-Fokus, CSELT, France Télécom R&D et Télécom Portugal.

message dans un autre format en cas d'échec et, en dernier lieu, notifier le problème à des experts du domaine.

Evolution du contrôle de la fourniture de services

L'introduction de nouvelles caractéristiques réseau telle que la variation de la bande passante dans l'UMTS nécessite d'adapter les services en fonction du contexte. Cela pourrait conduire à une résolution dynamique « en temps réel » des services alors qu'actuellement l'exécution des services est basée sur une résolution pré-câblée. Les propriétés suivantes actent en faveur d'une telle évolution.

La bande passante des mobiles UMTS varie en fonction de la vitesse de déplacement d'un utilisateur. L'optimisation de son exploitation nécessite d'adapter le service, par exemple le mode de restitution (vidéo, image ou texte), en fonction de l'usage et du type de terminal utilisé.

L'adaptation des services en fonction des données personnelles va s'intensifier, notamment à travers l'infrastructure PCS. Les données personnelles des clients mémorisées dans le réseau (carnet de coordonnées téléphoniques, etc.) sont amenées à être enrichies par l'apprentissage automatique des modes d'interaction des utilisateurs (appel fréquent de numéros depuis une localisation donnée, etc.). Ces données personnelles pourraient être utilisées et enrichies en dehors du *network home*, d'où l'intérêt d'une gestion individualisée et décentralisée des profils.

Le développement de nouveaux services de téléphonie a engendré une complexification des logiciels supportant *la logique de traitement*. Ces services sont souvent « composites » dans la mesure où il s'agit d'un agrégat de services plus élémentaires. En réponse à ces évolutions, les RI ont extrait des commutateurs des composants logiciels. Les composants logiciels supportant les services composites sont interfacés avec des composants logiciels de services plus élémentaires. Dans les cas où plusieurs compositions de services peuvent fournir un même service composite, il est possible d'envisager une résolution dynamique en fonction des services élémentaires disponibles sur un domaine.

L'infrastructure VHE, qui permet à un utilisateur où qu'il soit, y compris à l'étranger, de disposer de façon virtuelle des services propres à son opérateur, peut être réalisée de façon incrémentale, les services auxquels elle s'applique étant alors considérés les uns après les autres en fonction de la demande. Une alternative est d'anticiper ces adaptations et d'examiner les infrastructures logicielles nécessaires à l'adaptation de l'ensemble des services du VHE. L'introduction d'une résolution dynamique des services composites en fonction des ressources disponibles s'inscrit dans cette perspective. De façon analogue, l'interaction entre des composants de service issus de réseaux hétérogènes peut être abordée au cas par cas ou de manière plus globale. L'élaboration d'une couche abstraite masquant les spécificités des

différents réseaux est un moyen d'anticiper la convergence de services offerts par des réseaux hétérogènes. La réalisation de cette abstraction conduit à l'expression d'une ontologie du domaine définissant des concepts associés aux notions de service, d'allocation, de qualité, etc.

L'utilisation combinée de plusieurs composants de service peut entraîner des conflits de services, encore appelés *interactions de services*. Une interaction de services apparaît par exemple lorsqu'un utilisateur est abonné à un service permettant que son numéro ne soit pas connu et qu'il appelle un poste affichant le numéro de l'appelant. La modélisation du raisonnement sur l'intention est un moyen d'aborder la résolution d'interactions de services. Une solution est dans l'exemple précédent de considérer que l'intention de l'appelant n'est pas de masquer son identité mais de ne pas transmettre son numéro de téléphone et que l'intention de l'appelé est de connaître l'identité plus que le numéro de l'appelant. L'usage d'un service de télécommunication peut ainsi être dissocié du service et formalisé en terme d'intentions communicationnelles (informer, avertir, notifier, converser).

Intérêt des techniques agent

Dans ce contexte, les techniques agent s'inscrivent dans la continuité d'une approche componentielle. Un agent est un composant dont les interactions avec les autres composants portent sur la fourniture de services et sur la qualité des services. Cette interface est basée sur un langage commun à l'ensemble des composants. *A priori* ces composants ont la possibilité de communiquer par ce langage avec l'ensemble des agents de la plate-forme et des plates-formes interconnectées. Les agents sont généralement des composants dont les processus de traitement sont décrits et parfois conçus comme des processus de raisonnement sur des connaissances concernant les services et les ressources réseau disponibles. Au niveau des rôles fonctionnels, les agents sont souvent des composants avec des capacités de négociation sur la qualité de service. Les techniques agent constituent ainsi un moyen de conceptualiser la résolution dynamique de services composites à partir de la recherche et de la composition de services élémentaires. Cette approche est particulièrement intéressante lorsqu'il n'est pas possible d'anticiper la définition des futurs services. Un marché très ouvert peut engendrer une conception décentralisée des services qui rend ce type de prédiction très difficile, les composants logiciels de services étant conçus et administrés par différentes autorités. Il est par ailleurs possible d'envisager l'utilisation d'une ontologie de services pour automatiser des processus de négociation de fournitures de services. Les agents constituent un moyen d'uniformiser le traitement des profils utilisateurs et des composants de service. L'introduction d'agents pour représenter des assistants personnels prend tout son intérêt lorsque les comportements d'assistance varient suivant les usagers. Dans le cas contraire, un service global d'assistance dont les paramètres varient suivant les usagers est suffisant.

Problèmes

Les travaux de recherche sur l'application des SMA aux réseaux mobiles sont récents et ont donc été peu expérimentés. Les spécifications demeurent définies à un niveau abstrait ou sont déclinées sur des domaines restreints. Par ailleurs, il n'a pas été démontré que la coopération entre agents s'applique de façon opérationnelle à la combinaison dynamique de services et à la résolution d'interactions de services. Le développement d'une telle approche nécessiterait en outre la formulation d'une *ontologie des services de télécommunication*. Une approche générique telle que celle esquissée à partir des agents deviendra nécessaire dans ce contexte uniquement si les services couverts par le VHE et les possibilités de personnalisation offertes par le PCS se multiplient.

6.3.3. Application des SMA à la supervision et gestion de réseaux

La supervision de réseau, notamment la reconnaissance de pannes à partir d'observations locales d'alarmes peut être considérée comme une résolution distribuée des problèmes [PLU 96]. Dans ce domaine d'application, les agents désignent des composants logiciels contrôlant ou supervisant de façon décentralisée des ressources réseau. Ils sont utilisés pour développer des stratégies de coopération permettant de coordonner l'allocation ou la supervision de ressources dépendant de différentes autorités, ainsi que pour développer des stratégies de contrôle de la surcharge du réseau qui pourrait être engendrée par la signalisation liée aux nouveaux services.

Maquette d'un système de supervision de réseau

Une maquette a été réalisée à France Télécom (1993-1995) pour tester l'applicabilité des techniques agents à la supervision de réseaux [PLU 96] et examiner l'adéquation du langage AGENT0 [SHO 93] dans ce contexte. Dans cette maquette, l'organisation des agents est hiérarchique. Elle est calquée sur les flots de données lors du traitement des alarmes. La supervision est centralisée au niveau de l'agent superviseur. Celui-ci collecte des informations recueillies par les agents associés aux différents centres techniques (CT) et détecte d'éventuelles anomalies. Les agents des CT supervisent quant à eux des agents associés à des commutateurs. Ces derniers sont chargés d'envoyer des informations sur l'état des stations du commutateur à l'agent « centre technique » dont ils dépendent. Ils peuvent prendre l'initiative d'informer leur CT d'une « faute » en s'appuyant sur le fait que si un CT est intéressé par le changement d'état d'un commutateur, *a priori* il est intéressé par les fautes sur ce commutateur. Les comportements des agents sont décrits en termes d'engagements et d'actes de langage. Le langage AGENT0 est utilisé pour formaliser des procédures telles que « informer un agent de l'état d'une machine si cet état a été modifié et si l'agent avait demandé au préalable d'être informé de ce

type de fait ». Les agents ont une base de croyances où ils mémorisent des faits dont la valeur de vérité peut être incertaine. Certaines procédures engendrant des communications avec d'autres agents peuvent être activées en fonction des croyances d'un agent. Les scénarios de panne sont définis comme des séquences de messages. La reconnaissance de scénario est modélisée comme l'identification d'un message suivie de la génération d'un engagement à identifier la perception d'un message et ceci de façon itérative.

ACTS MARINER ET ACTS IMPACT

Le projet MARINER¹² a étudié l'intérêt des agents pour optimiser la gestion de la signalisation des réseaux. Les agents sont utilisés pour éviter qu'une congestion du réseau ne dégrade des services. Les facilités de communication entre agents ainsi que leurs capacités d'autonomie et d'adaptabilité sont exploitées pour accroître les performances du réseau intelligent en prenant en compte les problèmes de temps de réponse et de sécurité. L'interface entre l'architecture SMA et le réseau est supportée par une architecture RI de type SS7 qui est basée sur CORBA.

Le projet IMPACT¹³ a implémenté des stratégies de contrôle de qualité de service sur une plate-forme ATM à partir de techniques agent. L'allocation de la bande passante est considérée comme un des points de contrôle essentiel de la qualité de service. Les applications existantes communiquent à travers un commutateur pour demander l'établissement de nouvelles connexions. Chaque commutateur est interfacé avec un agent. Il peut par cet intermédiaire communiquer avec d'autres agents pour demander des ressources. L'agent est chargé de négocier les ressources et de comptabiliser le coût des communications.

Intérêts et problèmes

La description des comportements en terme d'engagements et de croyances apporte une lisibilité à l'opérateur sur les comportements des agents.

Bien que l'étude de l'application des agents à la gestion de réseau ait débutée dès le début des années 1990, les travaux sur ce domaine restent peu développés. Il existe des solutions alternatives fondées sur des approches distribuées [VU 01] pour traiter des problèmes dans lesquels l'organisation des agents est hiérarchique et où le contexte n'est pas un environnement ouvert. Ces solutions incluent des mécanismes de

12. Le projet ACTS MARINER (1998-2000) a été réalisé par : Teltec, Ericson, Université de Karlskobana, Imperial College, Telkom South Africa, IONA, IKV++ (<http://www.teltec.dcu.ie/mariner>).

13. Le projet ACTS IMPACT (1998-2000) a été réalisé par : Queen Mary Westfield College, ASPA, Flextel, University of Athen, Swisscom, Teltec, Tele Danmark (<http://www.acts-impact.org/>).

reconnaissance de scénarios plus dédiés et efficaces que ceux basés sur les langages agent. Les maquettes de supervision ont montré que certains comportements étaient communs à plusieurs types d'agents. Mais il s'agit en fait plutôt de facilités de calcul que de véritables attitudes comportementales de supervision.

6.4. SMA dans l'ingénierie logicielle des télécommunications

Dans les télécommunications, les agents sont souvent appréhendés exclusivement d'un point de vue service. Or, il est intéressant de les considérer également d'un point de vue génie logiciel. Cela permet, d'une part, d'aborder l'interopérabilité des agents à travers la mise en place de standards et, d'autre part, d'examiner l'apport des agents par rapport à des logiciels réalisés exclusivement pour la fourniture de services donnés.

6.4.1. FIPA : un standard d'interopérabilité agent

Les normes et standards dans les télécommunications ont d'abord porté sur les caractéristiques des signaux transmis puis, suite à l'évolution des télécommunications, sur le codage des données (ASN1, MPEG, etc.). Elles tentent désormais de s'appliquer aux couches logicielles. Dans le contexte des logiciels agent, l'intérêt majeur d'une standardisation est d'anticiper et d'encourager le développement de nouveaux services adaptés à un environnement ouvert.

Présentation de FIPA

FIPA est le premier consortium à mettre en œuvre un processus de standardisation autour des technologies agent. Le standard de facto KQML a été un préalable à cette standardisation. FIPA regroupe une cinquantaine de sociétés, des industriels des télécommunications (Alcatel, British Telecom, France Télécom, Siemens, NTT, etc.) des constructeurs informatiques (IBM, SUN, Motorola, Toshiba, Itachi, etc.), des industriels du domaine aérospatial (Boeing, Lockheed) et des universités (Imperial College, l'université du Maryland, etc.). La standardisation est basée sur l'intégration d'éléments techniques (des interfaces, des services d'administration, des protocoles et des langages). Des spécifications ont été publiées en 1997, 1998 et 2000.

Dans un contexte industriel concurrentiel, les détails d'architectures internes restent propriétaires et ne sont pas spécifiés par les standards. Cette caractéristique est difficilement conciliable avec la définition d'une sémantique d'interaction indépendante de protocoles donnés qui est de façon intrinsèque liée à un modèle d'agent donné. FIPA a de ce fait été confronté rapidement à une double contrainte : (1) ne pas imposer une architecture d'agents et (2) définir une sémantique par

rapport à un modèle d'agent donné. La sémantique des interactions est définie en référence à un modèle d'agent rationnel basé sur des modalités de croyances et d'intention. Les mécanismes d'interaction peuvent être *a priori* utilisés par des agents basés sur d'autres types d'architecture par exemple des agents procéduraux dont les comportements sont décrits par des protocoles exprimés sous la forme de graphes d'états.

Aperçu des spécifications produites par FIPA

Les spécifications de FIPA couvrent principalement les services d'administration des agents et la définition des communications entre agents. Une architecture permettant d'administrer les agents a été spécifiée en 1997 et 1998 et révisée en 2000. Trois types de services sont définis : le *Directory Facilitator* (DF), le *Communication Channel* (ACC) et l'*Agent Management System* (AMS). DF offre des fonctions d'annuaire et de nommage. Chaque ACC a une ou plusieurs interfaces avec des protocoles de transport. Les services et protocoles de transport prennent en compte des spécificités de différents domaines (internet, *wireless*). Les solutions requises sont par exemple IIOP et un codage sous forme de chaîne de caractères dans le contexte TCP/IP, WAP et un codage de type *bit-efficient* dans le domaine du sans fil. AMS définit les fonctions d'administration des agents.

Un élément fondamental de FIPA est la standardisation du langage de communication entre agents (ACL) et la recommandation d'un langage de contenu (SL). SL est un langage syntaxique permettant de dénoter et de décrire des objets, des actions, et des propositions. L'ACL est basé sur la définition formelle d'actes de communication entre agents. La plupart de ces actes sont utilisés dans la définition de protocoles de négociation, mais la signification des actes de langage n'est pas définie par cet usage. Le formalisme introduit dans FIPA a pour but d'exprimer sans ambiguïté la sémantique des actes de langage. Il n'est pas destiné à supporter une vérification automatique de la conformité des agents. Une extension d'UML (UAML) est en cours d'étude pour formaliser les protocoles d'interaction entre agents. Plusieurs bibliothèques de langage de contenus basées sur XML, CSP et KIF ont été développées. Une infrastructure de services de gestion d'ontologies permet aux agents de référencer et d'accéder à des ontologies.

Middleware FIPA

Les *middlewares* désignent des couches logicielles sur des architectures réparties qui sont interfacées avec les protocoles réseaux et qui offrent des fonctions de contrôle de services et de communication plus abstraites que celles fournies par des architectures spécifiques. Plusieurs *middlewares* « conformes » à FIPA sont en cours de réalisation. Des tests d'interopérabilité ont été conduits entre les plates-formes ASL, JADE [BEL 99], GRASSHOPPER [HÖF 99], MECCA [GER 01], FIPAOS, et

Comtec. De nouveaux tests sont réalisés depuis 2001 autour du programme Agent Cities de FIPA.

6.4.2. Méthodologie, ateliers de conception, de développement et de validation

La considération des agents d'un point de vue génie logiciel permet de dissocier le cycle de vie de l'agent logiciel du cycle de vie des services offerts par l'agent. Elle permet également d'envisager la perspective d'un déploiement industriel des techniques agent. La mise en place d'une méthode de conception agent s'inscrit dans cette perspective. En 1999, le consortium AgentLink a créé un groupe sur ce thème et un projet européen EURESCOM P907 a débuté autour de cet axe de recherche. Le projet CARISMA¹⁴ (conception, analyse et réalisation intégrée de systèmes multi-agents) avait anticipé l'évolution des recherches en ce sens [GER 98, YOO 99].

Projet CARISMA

Le projet CARISMA définit un environnement de conception, de génération de code et de vérification, appliqué à la réalisation de logiciels agents. Il est basé sur l'introduction des concepts d'agents, d'organisation, de communauté et de rôles. Les concepts de communauté, de rôles sont empruntés à ODP. L'environnement MADKIT qui introduit un modèle de conception graphique est utilisé pour procéder à des simulations. Cela permet une validation partielle avant de procéder à une vérification. Un formalisme de conception textuel a été défini en complément de la conception graphique pour générer le code automatiquement et pour faciliter la transition vers un formalisme de vérification. Il est dérivé des modèles BRIC et OfClass [YOO 99]. Le formalisme de conception textuel permet d'implémenter des réseaux de Petri représentant les agents. Les agents sont définis par des composants et leurs comportements internes par des capacités. Les composants peuvent hériter d'autres composants et inclure des sous-composants. Les opérations sont décrites en termes de transitions d'états. Une partie du comportement des agents n'est pas générée et vérifiée automatiquement, elle est décrite directement en Java. Le code généré est exécutable sur les environnements JNA [MER 98] et Agent Voyager. Des réseaux de Petri sont générés à partir de certains composants pour pouvoir procéder à des vérifications. Des règles de traductions ont été explicitées pour passer du formalisme de conception au formalisme de vérification. Ellesinstancient des classes, transforment les messages en flux et les ressources en états, initialisent les listes et les variables. Des composants sont ajoutés pour assurer des processus d'exclusion mutuelle. La complexité de la tâche de vérification est due entre autres à la taille de l'espace des états générés.

14. Le projet CARISMA (1996-1999) a été réalisé dans le cadre d'un contrat de recherche de France Télécom avec le laboratoire LIP6 de l'université Paris 6.

L'environnement CARISMA a été testé sur une version simplifiée du scénario d'assistance au voyage comprenant un agent utilisateur, un agent courtier et deux agents représentant des compagnies aériennes. Les composants définis dans le formalisme de conception correspondent aux différents agents et aux modules de gestion de tâches dans les protocoles de négociation. Ces composants sont décrits dans le formalisme de conception et réécrits dans le formalisme de vérification.

Une première vérification est basée sur le calcul de l'intervalle du nombre de jetons pouvant être dans chaque place. Cette vérification est faite à partir d'un outil de calcul de bornes basé sur des techniques de programmation linéaire. Les places dont le nombre de jetons n'est pas borné sont critiques. Il a été constaté ainsi qu'un agent qui ferait des appels à propositions de façon répétitive pourrait ne jamais examiner les offres faites en réponse à ces appels. Cela peut causer une incohérence dans le comportement de l'agent et engendrer l'explosion de la boîte de réception des messages correspondant aux offres.

Un deuxième type de vérification est basé sur l'identification des flots. Un flot est l'ensemble de places où circule un jeton donné. Cela permet, lorsqu'un flot couvre toutes les places d'un composant, de vérifier que la découpe componentielle est adéquate. Une troisième vérification considère des informations sur le graphe d'état. Elle permet de vérifier qu'il n'y a pas de nœud bloquant en dehors des nœuds terminaux. Le graphe d'état sert ensuite à vérifier des propriétés formulées en logique temporelle. Une des propositions vérifiées était : « Si un contrat a été initialisé alors le composant *contract-bidder* recevra une requête dans le canal CFP, ensuite le composant *contract-manager* pourra recevoir une réponse, etc. » Le nombre de chemins conduisant à une vérification de cette proposition est calculé. La vérification a montré qu'une partie de la proposition n'était couverte par aucun chemin ; cela est dû au fait que le *contract-manager* n'examinait pas de façon systématique les propositions reçues. Une stricte conformité à FIPA aurait également permis de progresser en ce sens puisque quand un agent envoie un CFP, il s'engage à procéder à une sélection et donc à une consultation des offres reçues en réponse. La vérification a en outre souligné un risque lié au fait que les protocoles Request et Query font appel au même canal Refuse. C'est une propriété intrinsèque à la définition des actes de langage FIPA. Le problème est dû à l'absence de composant modélisant l'unicité du sens des actes de langage.

Intérêts et limites du projet

L'intérêt majeur du projet est d'étudier simultanément les phases de conception, de simulation, de génération de code et de vérification. Le projet CARISMA a montré la possibilité d'utiliser un environnement de vérification basé sur des réseaux de Petri dans le contexte des systèmes multi-agents. Il a permis de cerner les limites d'une telle vérification (restriction des protocoles d'interaction et restriction de l'ouverture de l'environnement).

Le formalisme de conception a évolué au cours du projet pour lever des contraintes empêchant la vérification. Les modèles d'agents initiaux dont les comportements étaient complexes et parfois non déterministes ont été adaptés. Cela a conduit à situer le projet à la limite de la rupture avec le concept d'agent. Les actes de langage sont ramenés à des types de message. Par ailleurs une hypothèse forte du projet consiste à dissocier les comportements des agents des protocoles d'interaction. Cela permet notamment de restreindre la formalisation aux protocoles d'interaction. Cette hypothèse reste à valider.

Actions internationales sur le développement d'une méthodologie

La plupart des industriels de télécommunication ont désormais une expertise sur le domaine des techniques agent qui bien que développée dans différents projets semble converger. Il semble qu'elle soit basée sur une même approche méthodologique. Le projet EURESCOM P907¹⁵ a pour objectif de définir et de publier une méthodologie agent applicable au domaine des télécommunications. Il analyse les outils (*middleware*, architecture) qui ont été développés dans le cadre des précédents projets pour déterminer quels sont les aspects de cette méthode pouvant s'appuyer sur des outils existants.

6.5. Conclusion

Ce chapitre a présenté un état de l'art des travaux de recherche sur l'application des SMA au domaine des télécommunications. Une question importante d'un point de vue industriel est de déterminer dans quelles circonstances et à quelle échéance ces travaux pourraient aboutir à un déploiement industriel. Des éléments de réponse peuvent être apportés en précisant les rendez-vous des SMA avec l'émergence de marchés de services ouverts, l'implication des SMA dans l'innovation de services et le rôle des SMA dans l'anticipation de la convergence de services.

Les SMA sont difficiles à mettre en œuvre et ne présentent un intérêt que lorsque des solutions plus simples ne sont pas appropriées, c'est-à-dire dans le cas d'environnements complexes. Un tel environnement est caractérisé par un grand nombre d'acteurs, ou de composants, de larges possibilités d'interaction (chaque composant peut interagir avec un nombre important d'autres composants). Un moyen de calcul permettant d'évaluer cette complexité fait actuellement défaut. Le *web* est l'exemple type d'un tel environnement ouvert. L'intérêt des SMA dans le domaine des télécommunications me semble de ce fait étroitement lié à l'émergence d'un marché de services ouvert que ce soit pour des applications de commerce

15. Le projet EURESCOM P907 (1999-2001) est réalisé par : British Telecom, le CSELT, KPN, Telefonica, Broadcom, Belgacom, PT.

électronique ou des applications de négociation de services réseaux. L'émergence d'un marché ouvert pourrait être engendrée par une forte croissance de ces marchés et se produire vers 2005-2007. Le cas échéant, les techniques SMA pourraient être une des solutions techniques-clés si elles sont complètement opérationnelles à cette date.

Actuellement, des problèmes critiques demeurent sur les aspects sociaux des agents (définition de l'autorité, de la responsabilité, engagement vis-à-vis d'un tiers), sur la définition d'ontologies liées à la fourniture et à la négociation de services et sur le manque d'environnements logiciels permettant le déploiement de ces techniques. Les projets réalisés ont testé des services à base d'agents mais n'ont pas considéré l'ajout de nouveaux services au cours du cycle de vie de l'agent. Il nous semble que ces problèmes pourraient être résolus avant 2005 si les initiatives de stimulation du marché (FIPA, Agentlink) arrivent à dynamiser le marché des SMA. Dans l'hypothèse inverse, des techniques plus spécifiques à chaque domaine d'application seront utilisées.

Une démarche stratégique motrice actuellement dans le domaine des télécommunications est « l'innovation de services ». Elle a pour objectif principal de démarquer l'offre d'un opérateur des offres concurrentes. Les concepts recherchés par les unités de *marketing* pour développer des services innovants doivent apporter une forte valeur ajoutée et nécessiter la mise en œuvre de techniques peu complexes. Les services de personnalisation à base d'agents sur le *web* sont caractéristiques de ce type de services. Les SMA nécessitent pour leur part la mise en œuvre de techniques complexes. Leur utilisation dans le cadre du commerce électronique pourrait cependant être engendrée par l'explosion du marché et la nécessité de le structurer de façon dynamique.

Dans le domaine du contrôle de services de télécommunication, la problématique abordée par les SMA (et les agents mobiles) n'est à notre avis pas encore bien posée. Nous pensons que sur ce domaine d'application, l'intérêt majeur des SMA est dans la mise en place d'une couche de conception unifiée permettant par son niveau d'abstraction et sa focalisation sur l'interaction d'anticiper la convergence de services issus de réseaux hétérogènes (fixe, mobile, internet). Pour anticiper la convergence, les SMA devraient être appliqués à la conception des services dès à présent. Cela entraînerait un surcoût dans la conception de services et nécessiterait de convaincre l'ensemble des branches opérationnelles responsables des différents réseaux. Le retour sur investissements pourrait se produire vers 2005-2007 quand l'essor des mobiles de troisième génération entraînera une forte demande de services convergents. Dans ce contexte, l'approche multi-agent gagnerait à s'inscrire dans un *business plan*.

L'avenir des SMA dans les télécommunications est peut-être dans une voie peu explorée actuellement : la simulation. Le marché des télécommunications de par sa dynamique, que ce soit sur l'internet ou sur les mobiles, est difficile à analyser. L'utilisation de simulations multi-agents est appropriée à ce type de contexte [BEN 01]. Elles pourraient s'inscrire dans un contexte de recherche portant sur la compréhension des rationalités des acteurs et sur la modélisation comportementale des clients

6.6. Bibliographie

- [BEL 99] BELLIFEMINE F., POGGI A., RIMASSA G., « JADE – A FIPA compliant agent framework », *PAAM'99*, p. 97-109, 1999.
- [BEN 01] BENSARD L., BOURON T., « Multi-agent simulation of virtual consumer populations in a competitive market », *SCAI'01*, IOS Press, p. 31-43, 2001.
- [BOU 92] BOURON T., Structure de communication et d'organisation pour la coopération dans un univers multi-agents, Thèse de doctorat, Université Paris 6, 1992.
- [BOU 97] BOURON T., « Les agents dans l'Internet », *JFIADSMA'97*, 1997.
- [BOT 99] BOTHOREL C., THOMAS H., « A distributed Agent based platform for internet user communities », *PAAM'99*, p. 23-40, 1999.
- [CAM 98] CAMPS V., Vers une théorie de l'auto-organisation dans les systèmes multi-agents basée sur la coopération : application à la recherche d'information dans un système d'information répartie, Thèse de l'université Paul Sabatier, Toulouse, janvier 1998.
- [CAR 98] CARLA S. *et al.*, « The application of intelligent agents to the management of software problems in telecommunication », *IATA'98*, 1998.
- [CHI 98] CHIARIGLIONE L., « Agent technologies achieve maturity », *AgentLink news letter*, n° 1, p. 2-5, <http://www.fipa.org>, 1998.
- [DOO 97] DOORENBOS R.B., ETZIONI O., WELD D.S., « SHOPBOT, A scalable comparison-shopping agent for the world-wide web », *Autonomous Agent'97*, 1997.
- [FIN 97] FININ T., LABROU Y., « KQML as an agent communication language », dans *Software Agents*, AAAI/MIT Press, 1997.
- [FRI 97] FRIDMAN NOY N., HAFNER C.D., « The state of the art in ontology design: a survey and comparative review », *AI Magazine Fall*, p. 53-74, 1997.
- [GAL 00] GALANTE M.-P., LEICHSENRING A., MOREIRA M., SCARRONE E., STEGLICH S., AVELINE S., BOURON T., Evolution and application of enabling software technologies to mobile networks architectures, Projet Eurescom P810 deliverable 3, NT/FTR&D/6843, <http://www.eurescom.de/Public/Projects/P800-series/P810/p810.htm>, juin 2000.
- [GAR 98] GARIJO F. *et al.*, « Development of a Multiagent System for cooperative work with network negotiation capabilities », *IATA'98*, 1998.

- [GER 01] GERBER C., BAUER B., STEINER D., MotivPTA, Ressource Adaptation for a Scalable Agent Society in the Motiv-PTA Domain, <http://www.motiv.de/>, 2001 (à paraître).
- [GER 98] GERVAIS M.-P., DIAGNE A., « Enhancing Telecommunications Service Engineering with Mobile Agent Technology and Formal Methods », *IEEE Communications Magazine*, juillet 1998.
- [GUT 98] GUTTMAN R., MOUKAS A., MAES P., « Agent-mediated electronic commerce: a survey », *Knowledge Engineering review*, <http://ecommerce.media.mit.edu>, juin 1998.
- [HÖF 99] HÖFT M., MAGEDANZ T., et QUANTZ J., Grasshopper – The Agent platform, *AgentLink newsletter*, n° 3, p.7-10, <http://www.ikv.de/download/>, juillet 1999.
- [MAG 98] MAGEDANZ T., BREUGDST M., BUSSE I., COVACI S., Integrating Mobile Agent technology and Corba Middleware, *AgentLink news letter*, n°1, p. 5-9, <ftp://ftp.omg.org/pub/docs/orbos/97-10-05.pdf>, 1998.
- [MAG 99] MAGEDANZ T., ACTS Climate: An overview, *AgentLink news letter*, n° 2, p. 9-12, <http://www.fokus.gmd.de/cc/ima/climate/>, janvier 1999.
- [MER 98] MERLAT W., Adaptation dynamique de l'organisation dans les systèmes multi-agents. Application à la conception des systèmes coopératifs distribués et ouverts, Thèse de l'université Paris 6, décembre 1998.
- [PLU 96] PLU M., Expérimentation d'un modèle de programmation orientée agent en supervision de réseaux, NT/LAA/EIA/104, France Télécom/CNET, 1996.
- [SHO 93] SHOHAM Y., « Agent Oriented Programing », *Artificial Intelligence*, vol. 60 p. 51-92, 1993.
- [SMI 80] SMITH R.G., « The contract net protocol : high-level communication and control in a distributed problem solver », *IEEE transactions on Computers*, vol. C29, n° 12, p. 1104-113, 1980.
- [UMT 95] « UMTS Targets, System, Concept and Standardization in a global perspective », *IEEE Personal Communication*, vol. 2, n° 1, p. 18-27, février 1995.
- [VAE 99] VAN AEKEN F., Les systèmes multi-agents minimaux, Thèse de l'ENSIMAG, mars 1999.
- [VU 01] VU DUONG T., Découverte de chroniques à partir de journaux d'alarmes, application à la supervision de réseaux, Thèse de l'INP Toulouse, mars 2001.
- [WHI 97] WHITE J.E., « Telescript Technology: Mobile Agents, General Magic White Paper », *Software agents*, AAAI/MIT Press, 1997.
- [YOO 99] YOO M.-J., Une approche componentielle pour la modélisation d'agents coopératifs et leur validation, Thèse de l'université Paris 6, octobre 1999.

Chapitre 7

Systèmes multi-agents et écosystèmes

7.1. Introduction

Si, dans les prochaines années, une histoire des SMA en France est écrite, l'auteur situera certainement la naissance de cette approche et sa prime jeunesse dans le riche contexte du mouvement interdisciplinaire. En effet, contrairement aux Etats-Unis, où l'intelligence artificielle distribuée s'est développée sous l'égide des informaticiens indépendamment du domaine de la vie artificielle initié par les physiciens, en Europe et en France en particulier l'émergence de la communauté SMA est issue de la rencontre de chercheurs de plusieurs disciplines.

Dans le cadre général des sciences de la complexité, qui propose de reposer les questions scientifiques par l'étude des interactions entre entités élémentaires et de leurs organisations, les chercheurs en SMA vont participer à cette dynamique dans un double mouvement. D'une part, les SMA vont constituer une méthode pour catalyser la reformulation de certaines questions en sciences sociales ou en sciences de la nature. D'autre part, les chercheurs informaticiens vont puiser dans cette interaction les éléments pour mieux préciser les questions et proposer de nouveaux concepts dans leur discipline.

C'est ainsi, par exemple, que les chercheurs en SMA se posèrent dans un premier temps la question de la décision chez les agents, s'en allant interroger la psychologie cognitive et la théorie des jeux – pour rationaliser les stratégies qui anticipent la relation aux autres agents – avant d'aborder, dans un second temps, la question de

l'interaction entre les agents pour construire des sociétés artificielles, convoquant alors les sociologues – afin de construire les démarches d'interaction entre les individus et la société – ou bien les linguistes – afin de doter les agents de langages et d'organiser des protocoles de communication. Une des questions actuelles est celle de l'environnement des agents. Parmi les sciences mobilisées, l'écologie qui fait de l'environnement une de ses notions constitutives pourrait jouer un rôle primordial pour préciser les concepts et en conséquence développer des outils.

Dans le passé, les relations furent déjà fructueuses. Dans une acception très large, on trouve des références à l'écologie dès les premiers travaux sur les SMA. La métaphore sur la fourmilière fournit une illustration des plus évoquées pour vulgariser les notions d'agents réactifs et d'émergence et constitua le sujet des premières applications [COL 92, DRO 93]. Les « boids » de Craig Reynolds [REY 87] qui imitent les comportements de groupes d'oiseaux migrateurs semblent même précéder les notions de SMA ou de vie artificielle. S'ensuivirent de nombreux travaux sur les comportements animaux ou les sociétés animales. Aujourd'hui, les SMA sont aussi utilisés pour des applications dites environnementales, c'est-à-dire des applications qui impliquent des interactions entre dynamiques naturelles et dynamiques sociales.

L'ensemble des travaux dont il est fait état dans ce chapitre ont été classés suivant les communautés d'origine. Nous décrirons tout d'abord les raisons pour lesquelles l'écologie s'est intéressée à la modélisation de comportements d'agents autonomes et aux modèles qui ont été développés (*Individual Based Models*). Nous décrirons ensuite comment la communauté des SMA s'est enrichie des applications que des chercheurs en sciences de l'environnement lui ont proposées. Nous terminons par quelques éléments de discussion qui introduisent des perspectives de travaux communs.

7.2. Ecologie et modélisation de l'individu

7.2.1. De l'équilibre à l'organisation

Le domaine de l'écologie a subi ces dernières années quelques évolutions. Frontier [FRO 93] rappelle que le terme *écologie* a été créé par Hackel en 1866 pour désigner la science de l'habitat (du grec *oikos* : maison). Contrairement à l'éthologie qui s'intéresse essentiellement au comportement, l'écologie pose donc la relation au milieu comme constitutive de la discipline. Le mouvement historique retracé par Frontier rappelle une trajectoire déjà présentée. L'écologie a tout d'abord considéré les espèces dans leur singularité (autécologie) avant de s'intéresser aux interactions

telles que la compétition, la prédation, le mutualisme, entre les espèces (synécologie). Enfin, considérant que les organismes, populations¹ et peuplements modifient le milieu on parlera de l'écosystème, système d'interactions entre l'ensemble des populations. Se définissant ainsi, l'écologie se place dans le cadre de la théorie générale des systèmes que ses adeptes [LEM 89, MOR 77, VBE 68] présentent comme une théorie de l'interaction entre des parties d'où émergent des propriétés : le tout est plus que la somme des parties. Pour aller plus loin, face à la demande sociale de cette fin de siècle pour une recherche sur les questions d'environnement, l'écologie va répondre en intégrant le module socio-économique comme un sous-système pour étudier les interactions entre dynamiques sociales et dynamiques naturelles.

Si l'on peut ainsi positionner deux histoires sur la même trajectoire, la confrontation des idées véhiculées par ces deux champs de recherche ne se fera pas sans étincelles. En effet, si elle rejette le holisme, l'écologie, en s'interfécondant avec la systémique, a adopté les outils des automaticiens qui avaient fondé cette théorie : la dynamique des systèmes. Celle-ci décrit un système composé de compartiments ou modules (représentant des stocks, des agrégats) reliés entre eux par des flux de matière, d'énergie ou d'information et des contrôles. C'est ainsi que l'on intégrera les composantes écologiques et économiques en standardisant les flux entre modules [COS 93, ODU 57]. Le corpus théorique et les outils (Stella, Vensim, Simulink, entre autres) utilisés par cette démarche amènent à considérer les propriétés d'équilibre d'un système. Pour Uchmanski [UCH 96] cette perception (la nature est constante) représente les systèmes écologiques par des états stables définis comme des attracteurs que l'on cherchera à identifier par la dynamique des systèmes. Dans un essai sur la complexité des écosystèmes, Holling [HOL 73, HOL 87] définit trois concepts qui ont dominé la perception de la causalité dans les systèmes écologiques, avec pour conséquence de définir les éléments d'une gestion des écosystèmes. Le premier est centré sur la notion d'équilibre (*balance of nature*), le deuxième considère plusieurs états de stabilité (*nature engineered* ou *nature resilient*). Cette deuxième perception donne de l'importance aux dynamiques causées par la variabilité, par des événements qui apparaissent à de petites échelles.

1. *Population* : ensemble des individus d'une même espèce occupant un espace précisé.

Peuplement : ensemble fonctionnel de populations interconnectées ou susceptibles de l'être.

Biocénose : groupement d'êtres vivants correspondant pour sa composition, par le nombre des espèces et des individus, à certaines conditions moyennes du milieu. La dénomination d'une biocénose fait appel à un aspect particulier de son environnement. C'est donc l'ensemble de populations présentes dans un même biotope.

Biotope : aire géographique soumise à des conditions dont les dominantes sont homogènes. C'est l'espace occupé par la biocénose.

Ecosystème : c'est l'ensemble biocénose-biotope [BAR 92].

Plutôt qu'un seul équilibre, le système est caractérisé par un ensemble d'attracteurs. Le problème de gestion est de comprendre comment la variabilité endogène au système peut être contrôlée pour amener le système vers les attracteurs désirés. Le troisième point de vue est celui du changement organisationnel (*nature evolving*). Ce point de vue s'intéresse à la connectivité des éléments du système, à leur organisation² sur plusieurs échelles. Une référence des écologues sur ce dernier point de vue est la théorie de la hiérarchie exprimée par différents auteurs. La théorie de la hiérarchie est présentée par Allen et Starr [ALL 82] comme une théorie de l'observation de systèmes complexes. Ces systèmes complexes sont présentés comme les intermédiaires entre les systèmes à grands nombres pour lesquels une approche statistique est adaptée et les systèmes à petits nombres pour lesquels les approches mathématiques de type équations différentielles qui manipulent la dynamique de quelques populations sont pertinentes. Pour les auteurs, les systèmes intermédiaires restent opaques à moins qu'on les modélise en tant qu'organisations hiérarchiques [ALL 82, AUG 93]. Une organisation hiérarchique ne doit pas être vue comme un système de poupées gigognes mais plutôt comme des hiérarchies dynamiques enchevêtrées. Dans un récent ouvrage, Allen et Hokstra [ALL 92] expliquent que la compréhension d'un phénomène complexe en écologie implique de comprendre des interactions entre les différents niveaux : les organismes, les populations, les communautés, les écosystèmes, les paysages et les biomes : « *A part from organism and biosphere levels, there is plenty of room for entities from almost any type of ecological system to be contained within an entity belonging to any other class of the system.* » Les auteurs donnent l'exemple du phénomène de la rumination qui fait intervenir un écosystème sous le contrôle d'un organisme (le ruminant).

La vision systémique qui s'était en partie construite contre une approche réductionniste est aujourd'hui contestée par une écologie qui pose l'individu comme élément central et qui s'intéresse aux comportements et aux interactions entre ces individus. Cette nouvelle approche va développer son propre corps d'outils et de méthodes : les modèles basés sur l'individu (MBI) plus connus sous le terme anglo-saxon (IBM). Suivant les chercheurs et les applications, ces modèles seront utilisés dans le cadre de la deuxième ou de la troisième perception présentée par Holling. Dans un premier temps et pour nombre de chercheurs, l'introduction des niveaux individuels dans les modèles va répondre à une demande de prise en compte de la variabilité entre les individus. L'agrégation de tous les individus sous forme de population ou de cohorte était considérée comme trop réductrice. La représentation des individus, ou plus exactement de la variabilité entre individus, fut introduite afin

2. *Organisation* : en économie, mais le sens devrait être acceptable en écologie, c'est une procédure de coordination spécifique qui assure l'allocation des ressources selon des modalités propres [MEN 95]

de préserver une hétérogénéité considérée comme primordiale pour l'étude du système. Ceci ne remettait pas en cause les principes d'une approche cybernétique de la réalité dont les thèmes majeurs sont l'équilibre ou les dynamiques de rétablissement d'équilibre (résilience³). Dans un second temps et pour d'autres auteurs, l'entrée de l'individu dans l'organisation entraîne une démarcation fondamentale par rapport au point de vue cybernétique. Car l'individu est doté de caractéristiques uniques (référence à la génétique [JUD 94]) et il est porteur d'une histoire singulière [GRO 98]. Reprenant la classification de Holling, Villa [VIL 92] (voir tableau 7.1) invoque l'utilisation des nouvelles architectures informatiques et des nouveaux outils pour faire progresser ce nouveau point de vue en écologie.

Nombre d'applications présentées dans les ouvrages ou revues en écologie sous le sigle IBM pourraient être présentées comme des systèmes multi-agents. Nous présentons donc ici d'abord quelques principes et travaux marquants que l'on trouve dans la communauté de l'écologie.

	Point de vue dynamique	Point de vue organisationnel
Conceptualisation	Variables d'état	Processus/entités
Métaphores	Cybernétique	Parallélisme, automates cellulaires, systèmes d'apprentissage
Mécanismes	Centralisés	Distribués
Moyens d'analyse	Equations différentielles ou récurrentes	Simulation informatique
Comportements-clés	Equilibre, complexité dynamique	Auto-organisation, complexité structurelle
Organisation du système	Fixe, mononiveau	Variable, à plusieurs niveaux

Tableau 7.1. Deux systèmes d'interprétation écologiques, aboutissant à différentes notions de complexité (d'après [VIL 92])

3. La *résilience* est à la fois la capacité et la vitesse à laquelle un système écologique revient à son état après perturbation.

7.2.2. Les modèles basés sur l'individu

Les approches basées sur l'individu sont apparues à la fin des années quatre-vingt. L'article le plus souvent cité est celui de Huston [HUS 88]. Les deux arguments présentés par les auteurs pour le développement de cette approche sont, d'une part, la nécessaire prise en compte de l'individu dont la physiologie résulte d'une combinaison unique de la génétique et de l'environnement, et, d'autre part, le fait que chaque individu est situé et que ses interactions sont localisées. Ce plaidoyer fut suivi d'effets puisque de très nombreuses publications font référence à cette approche. Peu de temps après cette publication, Hogeweg et Hesper [HOG 90] publiaient un article du même type sur l'*individual oriented modelling*. Ces auteurs influenceront plutôt la communauté des chercheurs en vie artificielle. Puis, en 1990, se tint à Knoxville un congrès intitulé *Individual based models and approaches in ecology* dont les actes publiés constituent depuis l'ouvrage de base [DEA 92].

Un travail important fut mené par des chercheurs en écologie théorique. En 1994, Durrett et Levin publièrent un article dont le titre était : « L'importance d'être discret et spatial ». Dans cet article, les auteurs comparent quatre approches pour modéliser la dynamique de systèmes spatialement distribués : (i) le champ moyen (équations différentielles ordinaires) où chaque individu a une égale probabilité d'interagir avec les autres, (ii) les modèles dits *patches models* qui regroupent les individus sur un ensemble de lieux sans structure de connection, (iii) les modèles de réaction-diffusion où une population d'individus diffuse dans l'espace, et enfin (iv) les approches particulières où les individus sont discrétisés. Ils montrent les conditions dans lesquelles ces modèles sont équivalents ou non. Quelques autres travaux [MCC 93, WIL 93, WIL 96] ont continué dans cette voie de comparaison des résultats de modèles mathématiques et de simulations centrées sur l'individu. Au cours des années quatre-vingt-dix, un certain nombre de modèles basés sur l'individu ont été publiés. Certains modèles ne peuvent être reliés au domaine des systèmes multi-agents. Il s'agit des formalisations que l'on appelle *i-state* et *i-distribution models*. Ils reprennent les modèles démographiques classiques en introduisant une variabilité entre catégories d'âge ou de taille.

Les modèles réellement distribués sur l'individu que Maley et Caswell [MAL 93] appellent *i-state configuration models* ont le plus souvent fait appel à l'approche objet. On trouve ainsi les travaux suivants : [DER 98, LIU 98, ROE 91, SIL 93]. Plusieurs modèles sont des modèles de forêts [DEU 97]. On trouvera de très nombreuses applications en cours à partir du site *web* de C. Reynolds⁴. Il existe aussi des simulateurs dédiés [CAR 96, RIS 00]. L'application la plus marquante est sans doute le modèle ATLSS (*Across Trophic Level System Simulation*) qui cherche à simuler le

4. [Http://www.red3d.com/cwr/ibm.html](http://www.red3d.com/cwr/ibm.html).

fonctionnement écologique de la région des Everglades en Floride [ABO 95]. Ce modèle représente à la fois les facteurs abiotiques comme l'hydrologie, le feu, les ouragans, puis différents niveaux trophiques⁵. A l'intérieur de ces modèles, qui peuvent être des modèles mathématiques à compartiments, différentes populations d'animaux sont simulées (daims, félins) à l'aide de modèles IBM.

Dix ans après l'article fondateur de Huston, la communauté des chercheurs a récemment fait le point sur l'avancement des travaux. Plusieurs articles et un atelier de travail font une synthèse. Une part de cette synthèse discute les avancées en écologie, une autre part de ces synthèses fait le point sur la méthodologie et les questions informatiques.

En ce qui concerne les avancées en écologie, Grimm [GRI 99a, GRI 99b] classe l'utilisation des IBM suivant deux motivations. La première motivation est dite *pragmatique* : elle s'adresse à des problèmes qui ne pourraient pas être étudiés avec des modèles à variable d'état, comme par exemple expliquer une structure de taille de poissons [DEA 79] ou la dynamique d'une hiérarchie. Ce type d'application a donné lieu à beaucoup de modèles pour comprendre l'hétérogénéité d'une structure de population animale ou végétale. La deuxième motivation est dite *paradigmatique* : elle considère que les modèles à variables d'état sont pertinents mais montrent des déficiences pour étudier les questions classiques de l'écologie, à savoir la régulation, la stabilité, la résilience ou la densité-dépendance⁶. Dans le champ théorique, deux thèmes se dégagent. Le premier est la variabilité individuelle en tant que cause ou conséquence de la dynamique d'un système. On retrouve ici des questions d'interactions, soit entre les agents eux-mêmes, soit entre les agents et leur environnement. [UCH 96] et [LOM 99] mettent ainsi en avant la relation entre les caractéristiques individuelles et la distribution des ressources⁷ pour comprendre la

5. Les réseaux trophiques représentent l'organisation d'une chaîne alimentaire dont les niveaux correspondent en général à des espèces ; on étudie les flux entre niveaux et parfois la dynamique de cette organisation.

6. La *densité-dépendance* est une régulation qui provient du système (la densité) qui s'exprime au niveau des comportements et processus biologiques tels la reproduction, la croissance, la mortalité.

7. *Ressource*. Du point de vue de l'écologie, une ressource est définie comme la disponibilité de l'écosystème vis-à-vis de tel ou tel besoin précis de la population. Du point de vue de la géographie, une ressource est un moyen de créer de la richesse. Cette notion est relative, une ressource n'existe que perçue, nantie d'une valeur d'usage et si l'on est en mesure de l'exploiter ; sinon, elle n'est pas encore une ressource [BRU 92]. Du point de vue de l'économie, la notion de ressource est récente. L'analyse économique a d'abord utilisé les notions de capital, de biens et services et ce n'est qu'au début du XX^e siècle qu'elle s'est élargie au concept de ressource. Les ressources sont formées de facteurs de production, de la richesse naturelle et de capital accumulé [KEY 39].

régulation du système. Le deuxième thème qui se dégage est celui du descripteur de dynamique. Grimm propose de s'intéresser au concept de persistance. Si la stabilité suppose l'existence de un ou plusieurs équilibres, la persistance est la capacité d'un système à persister pendant un certain temps avec une certaine probabilité. On retrouve ici les idées développées par exemple par les mathématiciens [GAR 70]. Notons que, sur cette voie de recherche, les travaux menés en approche IBM conduisent souvent à des résultats aussi intéressants que surprenants⁸. Ainsi, pour Fryxel et Lindberg, qui ont fait un état de l'art de ce sujet, « *a growing body of work suggest that adaptive behavior has neither a routinely stabilising influence, nor a routinely destabilising influence* » [FRY 97]. Pour en terminer avec cette analyse partielle, Grimm tire un certain nombre de conclusions. Parmi celles-ci, l'idée d'un nécessaire couplage entre approches. L'approche ascendante qui utilise les IBM devrait être complétée d'une approche descendante classique en écologie qui définit les propriétés dynamiques au niveau de la population et fournit ainsi le cadre théorique auquel devraient se référer les IBM. Pour les auteurs, les approches qui utilisent les IBM, en s'appuyant sur l'utilisation de données, se rapprochent d'une science expérimentale et courent ainsi le risque d'une accumulation de cas sans tentative de généralisation. Le couplage avec des modèles agrégés de l'écologie théorique réduirait ce risque et enrichirait le débat.

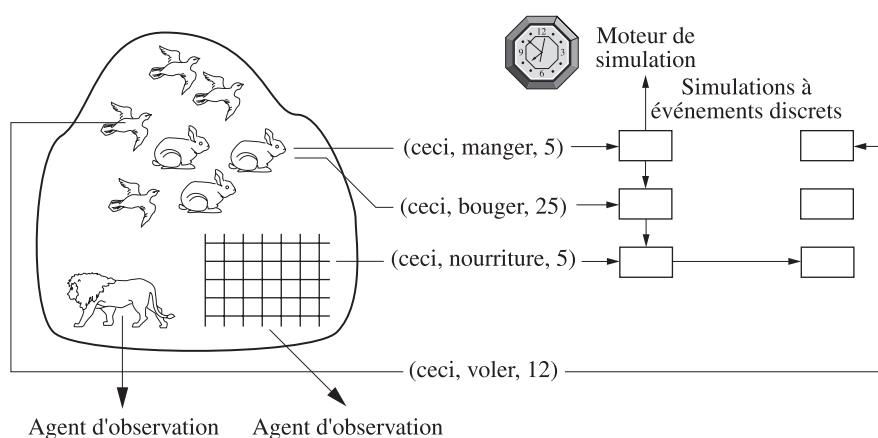


Figure 7.1. Architecture simplifiée d'Ecosim, composée d'un moteur de simulation basé sur des simulations à événements discrets, des modèles d'agents du domaine et des agents d'observation fournissant des indicateurs sur la dynamique d'agents ou d'ensembles d'agents

8. Pour une revue sur ce thème on pourra se reporter à la thèse de Shin [SHI 00].

Du point de vue informatique, un certain nombre de formalismes ont été utilisés, parmi lesquels on trouve les automates cellulaires [BAL 98, LET 99] ou les réseaux de Petri [BAK 00, GRO 98], mais surtout des modèles qui s'appuient sur l'approche objet. Dans un récent article Lorek et Sonnenchein [LOR 99] font une courte revue des outils disponibles. Ils listent Ecotalk, Hobo, Osiris et Swarm et leur propre outil (Ecosim). L'architecture d'Ecosim correspond à une architecture que les chercheurs en SMA reconnaîtront (voir figure 7.1).

Abott *et al.* [ABO 95] développent un modèle de dynamique des populations d'une espèce de daim en Floride au sein du projet ATLSS. Utilisant des machines parallèles ils discutent en particulier l'efficacité comparée de la distribution des portions d'espace ou des individus.

7.3. SMA et modélisation des écosystèmes

Parallèlement à la dynamique scientifique exposée ci-dessus et sans qu'il y ait beaucoup d'interactions – seuls quelques auteurs comme Hogeweg ou Bonabeau publient dans les deux communautés – de nombreux développements ont été effectués par des chercheurs informaticiens qui invoquent une référence à l'écologie. La revue que nous avons effectuée montre que dans une très grande majorité des cas, la méthode utilisée est celle de la simulation. Quelques recherches abordent la question de la résolution de problèmes, appelée *résolution collective de problèmes* [THE 94] dans le contexte des SMA. A partir de travaux en éthologie, sur les capacités de résolution de problèmes par des insectes sociaux [DEN 89], sont apparus de nombreux travaux dont les problématiques sont transposables à des problématiques logicielles ou en robotique. On trouvera une excellente revue dans [DRO 93] qui développe par ailleurs avec Ferber une approche originale nommée *l'éco-résolution*. Cette approche consiste à modéliser des agents qui doivent rechercher un état de satisfaction en fuyant un état de déplaisir. Un des exemples les plus connus est celui du taquin : en cherchant à se satisfaire individuellement, les agents accomplissent une tâche collective, celle de trouver une configuration spatiale. Née d'une collaboration entre informaticiens et éthologues, cette méthode pourrait servir pour des problèmes de gestion d'écosystèmes. Ainsi, les problèmes d'organisation spatiale d'un paysage à partir des différents rôles que l'on veut faire jouer à des portions de paysages en interaction. On en trouve un premier exemple dans le travail de Le Ber [LEB 99] sur l'organisation spatiale d'un territoire agricole : les occupations du sol sont représentées par des groupes d'agents qui cherchent à conquérir des parcelles afin de remplir un objectif de production. Les performances du système sont comparables à celle d'un algorithme de type *recuit simulé*. On peut classer dans la même catégorie les travaux de Baeijs [BAE 98] ou de Ferrand [FER 97]. Dans le contexte d'aménagement d'un espace, on laisse aux agents interactifs le soin de trouver une configuration spatiale qui optimise

un critère global. Une application est proposée qui concerne le tracé d'une ligne à haute tension dans un paysage de contraintes.

La majorité des travaux concerne la simulation d'«écologies». Les informaticiens concepteurs de systèmes multi-agents, ayant pour objectif de tester la fonctionnalité et l'efficacité de leur conception, simulent des sociétés d'agents situés, en relation ou non avec un environnement dynamique. Beaucoup de travaux ont traité à la gestion collective d'une ressource, quelques-uns à l'application de tâches collectives. Un grand nombre des publications qui font état d'un lien avec la biologie et l'écologie tiennent de la métaphore. Nous n'avons par exemple pas traité ici des travaux qui s'appliquent aux problèmes d'agents au sein de réseaux électroniques et qui font référence à des processus écologiques. Il s'agit d'un champ en émergence et le lecteur intéressé pourra se référer à [MAE 97]. Si la référence n'est aujourd'hui que métaphorique, l'intérêt des informaticiens devrait se confirmer pour le champ des recherches en écologie et en environnement car s'y pose la question de la survie de sociétés d'agents en relation avec un environnement commun. Les références à la tragédie des communs [HAR 68] commencent à apparaître ; les auteurs devraient bientôt se tourner vers toute la littérature sur la gestion de biens communs qui étudie l'interaction entre des processus sociaux et une ou des ressources communes. On y trouvera d'autres propositions de solutions que la solution économique (*market-oriented*) telle que Wellman par exemple la propose [WEL 96]. Si l'on considère ainsi les écosystèmes avec les hommes et les problèmes de gestion, Epstein et Axtell [EPS 96] étudient la structuration de réseaux et leur effet dans la gestion et la répartition des ressources. Dans un cadre plus appliqué, on remarque les travaux de Antona [ANT 98] et Van Parunak [VPA 98] sur la notion de filière (*supply chain*). Des ressources sont situées dans l'espace et des filières s'organisent pour collecter et écouler les produits. Dans ce cadre l'utilisation d'outils de gestion économique est proposée (quotas, taxes, permis...) [KOZ 99] et leur influence est testée en fonction du niveau de la filière sur lequel on les applique.

On trouvera dans les actes du colloque Smaget [FER 98] un large échantillon des recherches françaises dans le domaine des SMA en écologie et en environnement. Barreteau [BAR 00], Mathevet [MAT 00], Bousquet [BOU 01] entre autres, proposent des modélisations et des simulations qui impliquent des relations entre une ou des ressources naturelles, des agents qui individuellement peuvent aménager l'espace commun et agir sur la ressource commune et des jeux d'interactions entre les agents qui se coordonnent ou échangent. Un bon aperçu des travaux de la communauté américaine se trouve dans [KOH 00b].

La première partie de cette section propose une typologie des modèles dont la diversité est évoquée ci-dessus. Nous avons choisi de présenter les travaux selon le niveau sur lequel ils se concentrent principalement : (i) les agents et leur comportement,

(ii) les interactions entre les agents dans un environnement et enfin (iii) l'organisation de la société d'agents. Dans un deuxième temps, nous présentons les outils (plates-formes) de simulation couramment cités et utilisés pour le développement de ces modèles qui prétendent participer à la recherche en écologie. Enfin, nous opposons les SMA qui ont une vocation à apporter une contribution aux avancées théoriques et les SMA qui recherchent plutôt à modéliser un écosystème donné.

7.3.1. *Agents, environnement, interactions, organisation*

7.3.1.1. *Agents*

Bien que la notion de multi-agents implique l'interaction entre de nombreux agents, la bibliographie montre que la majorité des travaux accorde une très grande place aux mécanismes internes de l'agent, soit en termes de délibération pour des agents plutôt intentionnels, soit en termes d'adaptation pour des agents plutôt réactifs. Du point de vue des architectures rencontrées, on peut mettre en avant plusieurs types d'architectures. La plupart sont des architectures d'agents plutôt réactifs, mais dès lors que l'on considérera des actions humaines dans l'écosystème apparaîtront des architectures faisant plus de place à la délibération.

7.3.1.1.1. Architectures inspirées de la métaphore évolutionniste

De nombreuses applications des SMA à des écosystèmes proviennent de la communauté de la vie artificielle. Ce domaine qui cherche à comprendre la complexité dans une démarche ascendante utilise différentes techniques dont les systèmes multi-agents. Cherchant à comprendre la vie telle qu'elle aurait pu être plutôt que la vie telle qu'elle est [LAN 88], ce domaine utilise souvent des approches inspirées de la théorie de l'évolution pour comprendre l'adaptation. La méthode la plus connue est celle des algorithmes génétiques. Ils font appel à l'utilisation de ce que Oppenheimer appelle les *gènes numériques* [OPP 88]. Parmi les résultats de ses recherches sur les systèmes adaptatifs, Holland [HOL 75] a produit une classe d'algorithmes qui codent la solution potentielle d'un problème sous forme d'une suite de nombres ou d'une chaîne de caractères (chromosomes). L'algorithme attribue une valeur d'adaptation (*fitness*) à chaque chromosome. Périodiquement un processus de sélection, fonction de la *fitness*, fait évoluer la population en favorisant les plus adaptés. Des croisements et des mutations sont aussi opérés. De nombreuses applications ont été proposées avec de tels agents évolutifs ([LIN 94], par exemple). Plutôt que des interactions probabilistes, certains auteurs ont considéré chaque chromosome comme un agent situé. Le codage d'un chromosome au sein de l'agent consiste souvent à associer les différents comportements (déplacement, transport, communication, etc.) à des portions de chromosome (gènes) pour ainsi voir apparaître les combinaisons de comportements les plus adaptés. Collins et Jefferson [COL 92] présentent ainsi une des premières applications à des problèmes de *foraging* (recherche de nourriture). Krebs [KRE 96]

s'intéresse à l'émergence d'objectifs pour des agents qui doivent utiliser un environnement. Nishimura [NIS 97] présente une application de ces approches évolutionnaires et montre l'émergence de comportements collectifs de déplacements et de regroupement. Pour aller plus loin, J. Koza [KOZ 94] propose la notion de programmation génétique qui a pour but de faire évoluer des programmes informatiques. Quelques applications ont été proposées où l'on montre comment la structure de contrôle (le programme) d'un agent qui recherche de la nourriture évolue petit à petit pour s'adapter. Enfin, Holland a proposé une architecture d'agent inspirée de cette approche, qui donna lieu à la création de la plate-forme Echo [HRA 97].

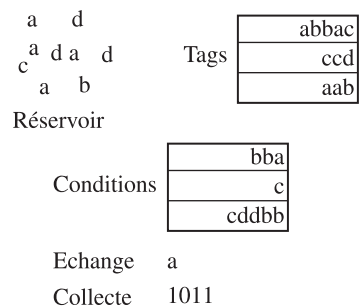


Figure 7.2. Architecture de l'agent Echo (voir explication dans le texte).
 Le génome résultant (qui code les tags, les conditions, l'échange et la collecte)
 est (abbac ccd aab bba c cddb a 1011)

Cette plate-forme a été utilisée pour diverses questions écologiques. L'agent de Echo possède une réserve de ressources (des lettres) prises dans l'environnement, un génome qui est composé de tags (qui exprime un phénotype, comme la couleur) et des conditions. Les gènes sont composés de ressources. Un gène particulier (collecte) spécifie les ressources que l'agent peut prendre dans l'environnement (le 1 spécifie que l'agent peut collecter la ressource, au contraire du 0 ; ainsi, dans le cas de l'agent Echo représenté à la figure 7.2, seule la ressource b ne peut être collectée). Six gènes spécifient les conditions et les tags pour l'attaque, la défense et l'accouplement. Un gène spécifie la ressource échangée. L'agent est répliqué lorsqu'il a acquis suffisamment de ressources pour dupliquer son chromosome. Des mutations peuvent apparaître. Les agents sur un même site vont interagir et en conséquence se transférer des ressources. Il y aura interaction entre les agents si le tag de l'un correspond à la condition de l'autre (pour le combat, l'échange ou l'accouplement). Le tag est la partie visible par les autres agents. Ce type d'architecture est utilisé pour des études en écologie théorique sur l'apparition de hiérarchies, sur l'émergence de mécanismes de compétition, la prédation, le mutualisme, etc.

7.3.1.1.2. Architectures pour les tâches compétitives

Différentes architectures ont été proposées pour représenter le choix que doit faire un agent. On trouvera de nombreuses connexions avec la robotique et la communauté qui travaille sur les animats [GUI 99]. S'inspirant de problèmes de recherche de nourriture, [TYR 93] propose une architecture où des systèmes comportementaux sont activés en parallèles. Poussant jusqu'au bout l'idée d'une intelligence sans représentation, certains auteurs représentent des agents dont les décisions s'appuient directement sur la perception de stimuli (béhaviorisme). Parmi celles-ci on distinguera entre autres une architecture MANTA proposée par Drogoul et Ferber [DRO 94] qui attribue un niveau d'activation pour chaque tâche concurrente. Ce niveau d'activation a_i d'une tâche i est calculé comme suit :

$$a_i(t) = \frac{w_i(t)}{\sum_{j=1}^n w_j(t)} x_i(t)$$

où w_i est le poids de la tâche et x_i l'intensité du stimulus. Une tâche est sélectionnée si a_i est supérieur à un seuil.

Pour sélectionner des tâches, R. Brooks a proposé l'architecture de subsomption [BRO 91]. Les activités sont représentées par différents niveaux. Le contrôle est basé sur les mécanismes d'inhibition et de suppression. Les niveaux supérieurs sont capables de supprimer les *inputs* des niveaux inférieurs et d'inhiber leurs *outputs*. Si elle est théoriquement toujours référencée et qu'elle paraît bien adaptée aux problèmes éthologiques, on trouve peu d'applications de cette architecture en écologie.

7.3.1.1.3. Architectures à base de réseaux neuronaux

L'accent est mis sur la capacité d'apprentissage des agents. La relation perception-action est modélisée par un réseau dont les connections évoluent. Collins et Jefferson dotent leurs fourmis de réseaux neuronaux pour leur permettre un apprentissage [COL 92]. La multiplicité des tâches à accomplir par un agent les conduit à construire plusieurs réseaux neuronaux (un pour apprendre à explorer, un pour apprendre à transporter). Jefferson *et al.* proposent aussi des réseaux neuronaux pour leurs agents de Genesys [JEF 91]. On trouve une application de ces méthodes dans la thèse de Laurent Dagorn [DAG 00], qui cherche à comprendre les déplacements de thons dans leur environnement. Le modèle Creature [GRA 98], qui est un jeu commercialisé, fait interagir le métabolisme de l'agent avec un réseau neuronal pour simuler le développement de l'agent : il acquiert un langage. Quelques applications couplent des réseaux neuronaux avec des algorithmes génétiques qui font évoluer le réseau.

7.3.1.1.4. *Fonctions paramétrées, les règles situées*

Le système le plus simple est celui constitué de bases de règles qui relient la perception à l'action. Elles sont souvent exprimées sous la forme de règles de production mais peuvent aussi correspondre à des fonctions. Par exemple, les travaux sur les vols d'oiseaux de Reynolds [REY 87] appliquent des calculs de vecteurs dans des champs de force résultant de l'attraction ou la répulsion d'autres agents. Ce type de modélisation où les agents représentent des particules élémentaires voit des développements importants dans différents domaines qui se rapprochent des modèles de dynamique des fluides : l'écoulement de l'eau [PER 97], les dynamiques de foules, le trafic urbain ou d'animaux [LAM 99, MEC 98, RAM 98]. Pour certaines applications dans le domaine de la gestion des ressources, on parle parfois d'*agro-écosystèmes*, on simule la décision d'agents économiquement rationnels. Afin de décider de leurs actions, ces agents vont utiliser des modèles issus de la recherche opérationnelle (calcul de gradient par exemple) ou de la micro-économie (comme la maximisation d'une utilité) qui cherchent à effectuer une optimisation sous contraintes. Il en est ainsi des travaux de [WEI 97], de [JAG 00] ou de [BAL 97] qui simulent l'optimisation d'un agriculteur au moyen de la programmation linéaire. D'autres applications (projet Fair-images) dotent leurs agents de méthodes de choix issus de l'analyse multicritère. L'agent peut même être représenté par un modèle de flux de matière ou d'énergie [GUE 99]. L'application de ces techniques pose souvent la question de l'articulation ou de l'arbitrage d'une rationalité individuelle pour laquelle de nombreuses méthodes sont donc disponibles avec une décision collective.

7.3.1.1.5 Architectures de type BDI (*Belief-Desire-Intention*)

Les agents cognitifs impliqués dans des écosystèmes apparaissent rapidement dans la communauté SMA [BOU 94, DOR 93]. Les objectifs, la représentation, l'engagement dans des actions individuelles ou collectives constituent les éléments des architectures proposées. Cependant la majorité des applications sur les problèmes écologiques engagent plutôt des agents simples dont on cherche à comprendre la coordination ou la relation au milieu. Il y a cependant une dimension cognitive qui paraît primordiale dans les systèmes écologiques : la représentation spatiale. Quelques premiers travaux [SAA 88] se sont intéressés à la représentation spatiale des animaux. La méthode la plus employée est la mémorisation de l'espace et des ressources au travers d'un ensemble d'objets correspondant à une carte mentale. Le travail précurseur de Folse [FOL 89] en est l'exemple type.

Des questions intéressantes peuvent apparaître sur des problèmes de coordination collective d'une représentation spatiale (ou d'une représentation collective de l'espace). Des questions apparaissent aujourd'hui sur des problèmes de partage à différentes échelles. Ainsi, comment un agent décide-t-il de quitter un lieu sur lequel les ressources sont rares et s'engage dans un long déplacement vers des zones supposées meilleures ? Cette question se pose en particulier pour les modélisations

de troupeaux [DUM 01]. Bah *et al.* [BAH 97] mettent en œuvre un calcul de zones viables à partir des représentations des lieux, dans le cas de la transhumance sahélienne.

7.3.1.2. Interactions et environnement

Dans le cas des SMA appliqués à l'écologie, on distingue trois grands types d'interactions : les interactions par communication entre agents, les interactions de type physique (pousser, manger) et les interactions médiées par l'environnement.

Le troisième type (médiation par l'environnement) est une réponse à ce que Kawata [KAW 94] définit comme l'une des deux questions majeures de la vie artificielle, à savoir la relation des organismes à leur environnement. Dans l'acception de l'écologie donnée au début, l'environnement ne peut pas être assimilé à l'ensemble des autres agents. Ce type d'interactions rejoint le concept d'externalité⁹ qu'utilisent les économistes. Le résultat de l'action d'un agent transforme l'environnement commun, ce qui rétroagit sur les autres agents. On parle d'externalités *positives* ou *négatives*. Dans les applications répertoriées, l'environnement peut être assimilé à l'espace et aux ressources. On trouve des recherches dans lesquelles l'état de l'environnement et son hétérogénéité sont utilisés comme support d'adaptation collective. On parlera de *coadaptation dynamique*. De nombreuses applications utilisent l'environnement comme un ensemble de signaux pour le déplacement ou pour la reproduction ou pour le choix de tâches. Il s'agit souvent d'algorithmes issus du concept d'*intelligence en essaim*, qui trouve son origine dans l'éthologie des insectes sociaux. Si les espèces servant de références sont souvent sélectionnées pour leur caractère social primitif [COR 94], c'est dans le but d'étudier l'émergence de l'organisation sociale (chez les espèces dites « eusociales », l'existence de castes plus ou moins spécialisées sous-tend cette organisation). Bourjot *et al.* [BOU 99a] proposent ainsi un modèle de simulation de l'activité de construction collective de toile chez les araignées. Dans tous les cas, l'idée forte est que « la structure de l'environnement et l'organisation du groupe d'agents se codéterminent mutuellement » [THE 94]. Nombreuses sont les applications, en particulier en France, où l'école réactive des SMA et le groupe de recherche en vie artificielle autour de Bonabeau et Theraulaz se sont consacrés à cette approche (stigmergie, coévolution, modèle du monde, etc.) [BON 99]. La confrontation interdisciplinaire a eu lieu, également avec les spécialistes des sciences de l'environnement qui, au cours des dernières années, se sont concentrés sur les interactions entre dynamiques naturelles et dynamiques sociales.

9. Même si plusieurs auteurs comme Arrow disent qu'il n'existe pas de définition claire du concept d'externalité, voici la proposition de Pigou [PIG 20] : « Action d'un agent économique procurant des avantages, ou des inconvénients, à d'autres agents économiques ne l'ayant pas choisi, sans que ceux-ci ne reçoivent ni ne payent de compensation à l'agent qui est à l'origine de l'action ».

Les interactions directes par échanges de message sont plutôt rares dans les applications en écologie. D'un point de vue métaphorique, quelques applications comme celle du modèle proie-prédateur étudié par Bouron ou Baray [BAR 98] sont citées. Les exemples les plus pertinents ont trait aux négociations, échanges de contrats, de biens, de services entre des agents qui représentent les humains qui font partie de l'écosystème. Ainsi, Franchesquin implémente le protocole de Sian pour simuler des négociations entre agriculteurs boliviens [FRA 95].

7.3.1.3. Organisation : le spatial et le social

L'organisation est-elle constitutive du SMA ou en est-elle le résultat ? Le débat sur l'émergence est moins simpliste aujourd'hui et s'organise autour d'une réflexion sur la circularité micro-macro [GIL 95]. A la fois produit, contexte et contrainte pour les agents, l'organisation est pourtant peu caractérisée. Si les structures d'agents ou les interactions sont catégorisées ou décrites, les organisations sont moins formalisées. On trouvera des éléments d'organisation dans les SMA dédiés à l'écologie sous deux aspects : le spatial et les réseaux.

La dimension spatiale est la plus fréquemment invoquée ; on décrit des organisations d'agents situés dans l'espace. La plupart des problèmes de recherche de nourriture passent par une organisation des agents et de leur environnement. On trouve aussi des recherches sur une question très importante en écologie : la régulation. Dans le cadre des relations d'agents avec leur milieu, on se pose la question de la quantité d'animaux qui peuvent vivre et se reproduire. Cela a pour directe conséquence de calculer les prélèvements possibles par la société et de penser les aménagements du milieu. Dans une vision cybernétique intégrative, et en s'inspirant de la démographie, de nombreuses recherches ont été effectuées sur le thème de la densité dépendance. Le concept de capacité d'accueil maximum K est la pierre de voûte de l'écologie du prélèvement. Les SMA ont permis de tester des hypothèses d'organisation qui sont des alternatives à la densité dépendance. Ainsi Le Page et Cury [LEP 96] testent la théorie dite *obstinate nature* : les agents tendent à se reproduire dans des conditions environnementales équivalentes à celles dans lesquelles ils sont nés. En croisant des comportements de déplacement liés à cette théorie et une structuration de l'espace, les auteurs décrivent une dynamique des populations régulée sans densité dépendance. Plusieurs recherches se tournent aujourd'hui vers la caractérisation des espaces dans lesquels se meuvent et se coordonnent les agents [PEP 99]. Le paysage peut lui-même être un SMA composé de différentes parties d'espace qui interagissent sur plusieurs échelles [LEP 99]. L'organisation de l'espace des agents et des ressources qui s'y trouvent peut aussi être le moteur d'une dynamique qui conduit à des mécanismes d'allocation des tâches. Drogoul [DRO 93] montre ainsi comment des agents (fourmis) peuvent se spécialiser dans différentes tâches, permettant ainsi le bon fonctionnement de la fourmilière.

Le deuxième type d'organisation que les SMA permettent d'étudier est celui de la structure des réseaux d'interactions. De nombreux travaux ont été effectués dans le domaine des réseaux trophiques et de la diversité des espèces. Les relations de causalité entre la stabilité d'un écosystème et son degré de connectivité ont ainsi été testées [BEG 90, MAY 73]. En figure 7.3, on voit des assemblages aléatoires d'espèces qui d'après May ont de faibles probabilités d'être dynamiquement stables. La compartimentalisation de communautés en sous-ensembles indépendants linéairement organisés donne une meilleure stabilité. Les contraintes thermodynamiques sur les flux d'énergie tendent à créer des structures pyramidales. Des règles simples empiriquement basées sur des flux d'énergie et des spécialisations trophiques, sont utilisées pour définir une organisation reflétant des contraintes « réalistes ». Des études de terrain [BRO 94] sur la réorganisation du réseau et les modifications du système, si l'on enlève une espèce du système par exemple, ont été réalisées. Les SMA semblent particulièrement adaptés pour simuler ces réseaux d'interactions et explorer les conséquences théoriques des différentes formes de réseau sur la dynamique de l'ensemble.

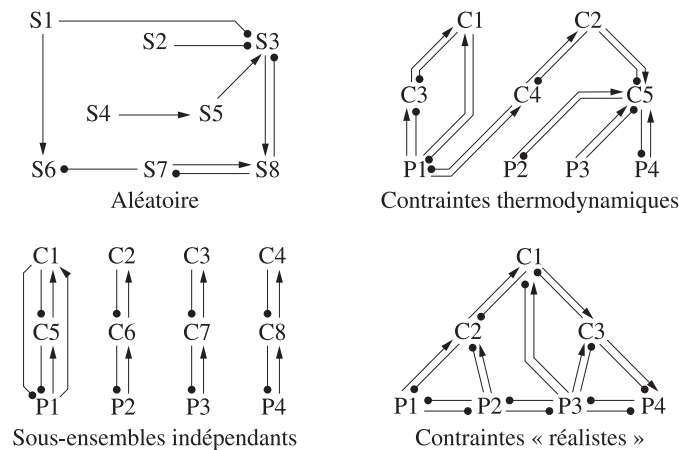


Figure 7.3. Réseaux hypothétiques d'interactions entre des espèces. Les flèches indiquent une influence positive, les ronds une influence négative (prédation, compétition, mutualisme ...).
S représente des espèces quelconques, *P* des producteurs, *C* des consommateurs
 (d'après [BRO 94])

Si l'on prend en compte la dimension humaine dans l'écosystème, certains chercheurs en sciences sociales modélisent et simulent des réseaux d'interactions entre agents pour analyser les effets de différentes rationalités et d'échanges. Ainsi,

Rouchier montre comment diverses hypothèses de relations entre des agents transhumants et sédentaires au Sahel donnent des dynamiques de la ressource très différentes [ROU 01].

Combiner la structure d'un réseau et sa disposition dans l'espace est la voie la plus riche. Les réseaux d'interactions entre espèces prennent d'autant plus d'importance s'ils se structurent dans l'espace. Dans un des modèles originels, Hogeweg [HOG 88] simule des agents, inspirés des insectes sociaux. Par le jeu des interactions entre les agents appelés TODO (comportement : faire les activités qui se présentent), et les agents DODOM (comportement : mettre en jeu les relations de dominance), des groupes sociaux se forment, un rythme se crée. L'influence de la structuration de l'espace sur la création de hiérarchies y est démontrée. Dans le même type de simulation, Doran étudie les réseaux sociaux qui se forment pour capturer des ressources situées dans l'espace [DOR 93]. Le travail porte sur une approche classique en BDI avec recrutement des agents pour accomplir une tâche. Des hiérarchies apparaissent et leur fonctionnalité est étudiée.

Réseaux trophiques, hiérarchies, filières, outils de régulation économique, enchères ; l'exploration des formes d'organisation des SMA est encore en cours mais on peut déjà imaginer les quelques formalisations qui vont se confirmer dans les années à venir pour compléter la structuration d'un SMA en s'ajoutant aux architectures d'agents, aux formes d'environnement et d'interactions. Le domaine général des institutions¹⁰ propose un cadre général pour étudier la gestion des biens communs et les mécanismes de régulation sociales, et devrait aussi fournir des éléments d'inspiration pour les SMA.

7.3.2. *Plates-formes*

Les applications présentées sont le plus souvent élaborées avec un langage orienté objet. Certaines d'entre elles utilisent des plates-formes. On peut classer ces dernières en trois types.

10. *Institution* : un ensemble de règles socio-économiques, mises en place dans des conditions historiques, sur lesquelles les individus ou les groupes d'individus n'ont guère de prise, pour l'essentiel, dans le court et moyen terme. Du point de vue économique, ces règles visent à définir les conditions dans lesquelles les choix, individuels ou collectifs d'allocation et d'utilisation des ressources pourront s'effectuer. Elles contribuent à définir les conditions sociohistoriques dans lesquelles les mécanismes de coordination peuvent s'instaurer [MEN 95].

Les plates-formes génériques : aujourd'hui, quelques-unes d'entre elles sont régulièrement citées dans des applications environnementales. Swarm [MIN 96] est l'outil privilégié de la communauté américaine et des chercheurs en vie artificielle, ainsi que la plate-forme Echo déjà décrite ci-dessus. De nombreuses applications ont été développées à partir de Swarm qui existe aujourd'hui en plusieurs langages (Java, Objective-C). Des outils génériques ont même été créés à partir de Swarm (voir ci-dessous). La plate-forme Geamas [MAR 97] a été utilisée pour quelques applications sur des écosystèmes. Alors que la plate-forme MadKit implémente la notion de rôle [GUT 97], Geamas propose une implémentation du groupe.

Les plates-formes orientées écosystème : ce type de plates-formes, parmi lesquelles on peut ranger l'outil Ecosim déjà cité ou encore les plates-formes Sugarscape ou Cormas [BOU 98], proposent des utilitaires pour simuler des écosystèmes ou des problèmes de gestion des ressources. Grilles spatiales, utilitaires de simulation pour méthodes de type Monte-Carlo, liens vers des logiciels (SIG, bases de données) font de ces outils des ateliers pour l'implémentation de divers systèmes écologiques. Des algorithmes ou des structures sont prévus pour implémenter le lien des agents avec leur environnement et on trouve aussi des éléments pour l'organisation de sociétés d'agents (marchés, enchères, mécanismes de prédation, etc.).

Les plate-formes dédiées : ces outils s'intéressent à des types d'applications plus précis. Ainsi Manta [DRO 93] s'intéresse aux problèmes de fourragement ou d'allocation des tâches dans une société d'insectes, Arborscape [SAV 00] modélise les dynamiques forestières en insistant sur la diversité, BacSim [KRE 98] modélise les dynamiques microbiologiques. Mobydic propose de modéliser les dynamiques des peuplements de poissons [GIN 98]. L'utilisateur bénéficiera de la réflexion sur les dimensions, sur des relations de type prédation ou compétition, sur les fonctions biologiques classiques (mortalité, croissance, etc.).

7.3.3. Objectif : propriétés génériques ou représentation des connaissances ?

Dans le champ de l'écologie, on pourra trouver des recherches sur les dynamiques de réseaux trophiques ou sur les relations entre des comportements et des distributions statistiques. Ce champ peut s'organiser autour d'une opposition méthodologique entre l'approche qui recherche les propriétés génériques et une approche plus inductive. Ces deux approches peuvent être utilisées pour étudier des propriétés qui s'expriment sous la forme de distributions ou d'indicateurs [HRA 97], des configurations spatiales, des formes de réseau.

Dans le premier cas des utilisations possibles de ces simulations est en concordance avec les principes de la vie artificielle : « La vie telle qu'elle pourrait

être plutôt que la vie telle qu'elle est » [LAN 88]. Le modélisateur met en place des mécanismes et observe les réponses émergentes. Eventuellement, ces formes existent. La raison de ces recherches vient des résultats et de la démarche de la physique [WEI 91a, WEI 91b] : on s'intéresse aux transitions entre phases d'un système. Il s'agit alors de construire des modèles très simples d'interaction et rechercher les coefficients critiques qui caractérisent les transitions. On suppose que le modèle et le système étudié appartiennent à la même classe d'universalité dont on a ainsi décrit les propriétés qualitatives.

Une autre utilisation, plus empirique, vient plutôt de la communauté des modélisateurs issus des sciences de la vie et de la société directement ou indirectement impliqués dans des problèmes de gestion des ressources. L'idée sous-jacente, qui est de faire un système qui se comporte comme la réalité, est toujours présente avec comme objectif de se servir du simulateur pour se poser la question « et si... ? » Cette adaptation du modèle à la réalité n'est pas destinée à faire du modèle un outil de prédiction mais plutôt à comprendre des dynamiques existantes ou ayant existé. Les auteurs cherchent le comportement, identifient les paramètres pour proposer non pas une explication mais une simulation des observations sur la réalité : l'hypothèse testée permet de simuler ces observations mais d'autres hypothèses pourraient aussi simuler cette réalité. Cette méthode est par exemple employée en archéologie [KOH 96] ou en histoire [KOH 00a]. Une application est celle de Dean [DEA 00] qui reconstituent l'histoire des indiens Anasazi et proposent par simulation des scénarios qui discutent les déplacements de populations en fonction des crises environnementales. Quatre modèles de réseaux trophiques sont présentés ci-dessous, illustrant les deux démarches.

Dans le cas des réseaux trophiques, [SCH 97] ont utilisé le modèle Echo qu'ils ont un peu transformé pour donner l'environnement de modélisation Gecko. Pour cela et par rapport à l'agent présenté à la figure 7.2, ils ont doté chaque agent d'une capacité de transformation : un agent peut transformer une ressource *a* en ressource *b* avec un coût associé. D'autre part, les agents prennent une autonomie supérieure car ils sont capables de faire un choix parmi les ressources qu'ils consomment. Le but de l'article est de tester les relations trophiques entre différents producteurs et des agents hétérotrophes. Les résultats montrent la sensibilité du système tant à cette capacité de choix des agents qu'au coût métabolique de la transformation. De même, [LIN 94] construisent des réseaux trophiques à partir des interactions entre des agents qui représentent des stratégies codées sous la forme de génomes. Des flux d'énergie sont simulés en prenant en compte la transformation énergétique au long du réseau. Les interactions sont représentées au moyen du dilemme du prisonnier. Après avoir observé l'apparition de réseaux trophiques « réalistes » en termes de ratio de productivité ou d'énergie, les auteurs introduisent des agents qui choisissent leurs interactions. Les auteurs voient alors apparaître, dans un processus de coadaptation, des stratégies nouvelles telles le camouflage.

Le même type de question s'est posé de façon très appliquée dans le domaine des pêches. Deux expériences menées à l'IRD cherchent à faire le lien entre des dynamiques de prédation et de compétition et des indicateurs écosystémiques. Dans une première expérience, un réseau trophique composé de trois espèces de poissons dont les comportements sont censés représenter la diversité des stratégies rencontrées sur le fleuve Niger a été simulé [BOU 94].

L'environnement est un système fleuve-plaine inondable représenté par plusieurs milieux qui procurent une offre de nourriture variable au cours du temps. Les agents qui sont implémentés présentent des comportements qui sont l'expression de mécanismes d'adaptation (on parle de stratégies adaptatives) : différents types de reproduction, de mouvements dans l'espace et dans le temps. Une espèce consomme le plancton offert par la crue. La seconde est hétérotrophe : elle consomme du plancton ou des petits poissons. La troisième est un pur prédateur. Les agents de ce système ne sont pas des modèles de l'individu mais plutôt des modèles de groupes. En partant de ce réseau trophique, on simule une pression d'exploitation (intensité de pêche) croissante. On peut alors observer l'impact de cette pression sur le réseau trophique, avec notamment la raréfaction d'espèces, ainsi que la réponse globale en termes de quantité de captures : on montre que cette réponse prend une forme de plateau, traduisant la résistance de l'écosystème à ce stress qu'est l'exploitation halieutique. Or cette forme de plateau est une forme généralement connue en écologie des ressources [WEL 89] voire même en tant que forme-réponse de tout organisme soumis à un stress. On passe donc de la connaissance sur les comportements des espèces à des caractéristiques de la dynamique du système.

Une autre expérience du même type est celle de Shin [SHI 00]. Des agents, dont la modélisation s'appuie sur des bases de données, situés sur une grille spatiale entrent en interaction de prédation. Le résultat de ces multiples interactions est observé à un travers un indicateur global, le spectre de taille, classiquement utilisé par les gestionnaires pour analyser la situation biologique.

Dans le cas de ces deux expériences il s'agit de faire le lien entre des données à différents niveaux : des comportements et des interactions au niveau micro et des formes (*pattern*) observée à l'échelle de l'écosystème.

7.4. Quelques éléments de discussion

Nous avons choisi dans cette présentation de faire un état de l'art en séparant les recherches sur les IBM et sur les SMA. La frontière n'est bien sûr pas étanche et plusieurs applications pourraient être revendiquées par les deux communautés. Cependant, globalement, il semble que les questions abordées soient différentes et que leurs perspectives soient complémentaires.

Dans le propos d'étape effectué par les chercheurs de la communauté IBM différentes conclusions sont tirées. L'idée générale est que, après quelques années d'innovations enrichissantes, la recherche doit passer par une phase de consolidation. Cette consolidation doit s'appliquer tout d'abord à la méthode. Les biologistes informaticiens doivent redevenir modélisateurs. Comment exprimer les résultats d'un modèle ? Comment présenter sa structure ? Il s'agit de réappliquer les étapes d'une science expérimentale. La consolidation est aussi théorique. Le conseil des auteurs est de prendre garde à l'accumulation de cas particuliers, de modèles trop appliqués. Il est proposé de favoriser les modèles proches de la théorie, exprimés en général par des modèles mathématiques, qui ont fait l'effort depuis quelques années de prendre en compte non seulement le discret et le spatial mais aussi l'individu. Cette discussion et ces recommandations viennent des représentations et des objectifs de ceux qui développent des IBM. Pour certains, ces derniers représentent une approche pour intégrer la variabilité interindividuelle dans des approches conventionnelles sur les écosystèmes. Il est alors temps de consolider les apports pour les intégrer aux démarches sur l'écosystème. Pour d'autres ces méthodes marquent une rupture dans la pensée. Les outils développés correspondent à une approche différente, à une rupture avec la systémique, vers un néo-réductionnisme qui s'appuie sur les nouvelles avancées de l'écologie comportementale ou sur une théorie de la hiérarchie qui cherche à comprendre les phénomènes écologiques à travers des interactions entre différents niveaux. Pour ces deux types de questions, il nous semble qu'à quelques exceptions près les approches IBM n'ont pas exploré suffisamment loin et que les SMA sont intéressants car ils apportent une formulation non seulement de l'autonomie de l'individu mais aussi et surtout des formes d'organisation. Faire un SMA sur une application écologique, c'est expliciter les relations des agents avec leur environnement, les protocoles d'interaction entre des agents qui représentent des niveaux d'agrégation différents.

Réciproquement, les intérêts d'applications en écologie sont nombreux pour la communauté des informaticiens. Concernant le domaine de la vie artificielle Bonabeau [BON 94] en désigne plusieurs dont la conception d'algorithmes et la théorisation. On y ajoutera ici un développement sur l'utilisation des SMA pour l'action.

La métaphore pour concevoir des algorithmes. Les relations modélisées lors d'une application servent pour conceptualiser des algorithmes, des protocoles, des systèmes de visualisation [HUT 00], voire des architectures. Les algorithmes génétiques, l'éco-résolution sont des premiers exemples, mais bien d'autres devraient apparaître au moment où les SMA se posent des questions de partage de ressources communes et au moment où la mode écologiste et environnementaliste procure un présupposé favorable à tout label environnemental apposé à une nouveauté computationnelle.

La théorisation. On retrouve ici les éléments de la discussion sur les IBM : les SMA peuvent être créés à partir des modèles mathématiques dans le but de les complexifier ou bien il est possible d'utiliser les SMA pour développer de nouvelles constructions théoriques. [CAZ 00, KEI 97, VPA 98, WEI 97] entre autres ont proposé les premiers travaux qui confrontent un SMA à des modèles mathématiques existants ou créés à l'occasion d'une application. En général le modèle mathématique¹¹ est d'abord élaboré (les méthodes de la physique statistique sont très utilisées), ses propriétés sont étudiées, puis les SMA permettent d'introduire de l'hétérogénéité, de l'anisotropie, des histoires locales. Mais les modèles SMA s'avèrent aussi de très riches méthodes pour élaborer directement de nouvelles théories, « *more theory building than modelling* » [DOR 93]. Ils doivent ainsi participer à des efforts de théorisation des relations. Cette théorisation n'interdit pas une confrontation avec les données de terrain. A côté de l'approche vie artificielle, s'est développée l'idée du laboratoire virtuel. Il s'agit de construire un modèle du monde et d'observer sa dynamique à travers des indicateurs élaborés selon les mêmes protocoles que ceux employés pour observer le réel. Des méthodes de validation, classiques en simulation [HIL 95], peuvent même être utilisées.

L'action. La vocation d'un modèle est le plus souvent de servir pour l'aide à la décision. La jeunesse des modèles SMA les a, pour un temps, confinés dans des domaines de recherche et leurs résultats n'ont pour l'instant que peu été confrontés à des problèmes de décision. Il est temps de définir des méthodes pour utiliser les SMA. Les modèles peuvent être utilisés pour proposer des solutions. Il est vraisemblable que le domaine de la résolution distribuée trouvera de nombreuses applications pour proposer des solutions de configuration d'un espace par exemple. Mais la simulation peut, elle aussi, être utilisée et participer à des processus de décision. Ainsi le travail de Gimblett *et al.* [GIM 98], qui ont proposé des aménagements d'un parc naturel pour éviter que se croisent des usagers concurrents (VTT, marcheurs, Jeeps) en simulant le déplacement et le champ de vision des agents. D'autres méthodes sont proposées telles la modélisation d'accompagnement [BOU 99b]. Cette méthode propose d'utiliser les SMA dans les problèmes de gestion des biens communs dans une démarche constructiviste avec les acteurs du système. Le modèle devient une représentation partagée et peut devenir, dans le temps du processus social, un outil pour une coadaptation dynamique entre un ou plusieurs groupes sociaux et leur environnement. Une démarche circulaire de présentation des modèles et de construction des modèles avec les acteurs a été proposée [BOU 99b] et testée sur différents terrains. La méthode utilisée est celle

11. On remarque qu'aujourd'hui c'est le modèle mathématique lui-même qui est appelé modèle multi-agents [JAN 00]. Ces modèles prennent en compte la multiplicité des interactions en général au moyen de calculs probabilistes, mais ne tiennent pas compte de l'unicité de l'agent, de son autonomie.

des jeux de rôles [BAR 99]. Le jeu de rôle peut être utilisé pour présenter un SMA ou pour le construire avec les acteurs : une modélisation *bottom-up* pour un processus de décision *bottom-up*.

7.5. Conclusion

Depuis le milieu des années 1980, différentes communautés de recherche travaillent sur les modèles distribués et les écosystèmes. Des écologues ont introduit la notion d'individu dans leurs modèles de dynamique pour mieux comprendre le fonctionnement des écosystèmes. Des informaticiens ont développé des modèles d'écosystèmes à la fois pour trouver de nouveaux modèles conceptuels de comportement et d'interaction et pour tester les architectures qu'ils avaient imaginées. Après des années enthousiasmantes dont quelques étapes et résultats sont relatés dans cet article, il semblerait que les chercheurs impliqués ressentent aujourd'hui une baisse de l'enrichissement. Les uns se demandent quel fut le réel apport de la prise en compte de l'individu, les autres se demandent comment éviter l'enlisement dans l'application. Ces deux mondes se sont sans doute moins rencontrés qu'ils ne le croient et un nouveau pas devrait être franchi dans les années à venir. Les chercheurs en écologie peuvent utiliser les systèmes multi-agents pour aller plus loin que la prise en compte de l'individu et étudier plus et mieux les formes d'organisation (réseaux, hiérarchies), les interactions entre différentes échelles. Les informaticiens peuvent quant à eux bénéficier des connaissances des écologues, pas seulement sur le comportement des individus, mais surtout sur les systèmes d'interaction entre des agents et leur environnement (relations entre des organisations et des propriétés de stabilité, résilience, résistance). Si chaque communauté pourra encore bénéficier de l'autre, c'est ensemble que des progrès seront effectués sur les méthodologies pour l'usage de ces modèles : méthodes d'observation d'un monde artificiel, expérimentation, résolution de problèmes, formes de participation au processus de prise de décision.

7.6. Bibliographie

- [ABO 95] ABO T. C., BERRY M.W. *et al.*, Computational models of white tailed deer in the Florida everglades, UTK-CS, 1995 (<http://www.cs.utk.edu/~berry/deer/deer.html>).
- [ALL 92] ALLEN T.F.H., HOKSTRA T.W., *Toward a unified ecology*, Columbia University Press, 1992.
- [ALL 82] ALLEN T.F.H., STARR T.B., *Hierarchy: perspectives for ecological complexity*, University of Chicago Press, 1982.
- [ANT 98] ANTONA M., BOUSQUET F. *et al.*, « Economic Theory of Renewable Resource Management: a Multi-Agent System Approach », dans J.S. Sichman, R. Conte et N.

- Gilbert (dir), *Multi-Agent Systems and Agent-Based Simulation*, Paris, France, Springer-Verlag, 1, p. 61-78, 1998.
- [AUG 93] AUGER P., « Quelques outils mathématiques pour l'analyse hiérarchique », dans P. Auger, J. Baudry et F. Fournier (dir), *Hiérarchies et échelles en écologie*, Naturalia Publications, 1993.
- [BAE 98] BAEJS C., Fonctionnalité émergente dans une société d'agents autonomes. Etude des aspects organisationnels dans les multi-agents réactifs, Thèse, INP Grenoble, 1998.
- [BAH 97] BAH A., Simulation Multi-Agents des modes d'utilisation des ressources en propriété commune ; le cas de la mobilité pastorale en zone intertropicale sèche, Ecole Supérieure Polytechnique de Dakar, 1997.
- [BAK 00] BAKAM I., KORDON F., LE PAGE C., BOUSQUET F., *Formalization of a spatialized multi-agent system using coloured Petri nets for the study of a hunting management system*, Springer-Verlag, LNCS, 2000.
- [BAL 97] BALMANN A., « Farm based modelling of regional structural change. A cellular automata approach », *European review of agricultural economics* vol. 24, n° 1, 1997.
- [BAL 98] BALTZER H., BRAUN P., *et al.* « Cellular automata models for vegetation dynamics », *Ecological modelling*, 107, p. 113-125, 1998.
- [BAR 98] BARAY C., « Effects of population size upon emergent group behavior », *Complexity international*, n° 6, 1998.
- [BAR 92] BARBAULT R., *Ecologie des peuplements, Structure, Dynamique et Evolution*, Masson, 1992.
- [BAR 99] BARRETEAU O., BOUSQUET F., « Jeux de rôles et validation de systèmes multi-agents », *Ingénierie des systèmes multi-agents JFIADSMASMA'99*, Hermès, p. 67-80, 1999.
- [BAR 00] BARRETEAU O., BOUSQUET F., « SHADOC: a Multi-Agent Model to tackle viability of irrigated systems », *Annals of Operations Research*, n° 94, p. 139-162, 2000.
- [BEG 90] BEGON M., HARPER J. *et al.*, *Ecology. Individuals, Populations and Communities*, Blackwell Scientific Editions, 1990.
- [BON 94] BONABEAU E., THERAULAZ G., « Why do we need artificial life? », *Artificial life* n° 1, p. 303-325, 1994.
- [BON 99] BONABEAU E., DORIGO M. *et al.*, *Swarm Intelligence. From natural to artificial systems*, Oxford University Press, 1999.
- [BOU 94] BOUSQUET F., « Distributed artificial intelligence and object-oriented modelling of a fishery », *Mathematical Computer Modelling*, n° 2018, p. 97-107, 1994.
- [BOU 98] BOUSQUET F., BAKAM I. *et al.*, *Cormas: common-pool resources and multi-agent Systems*, Springer-Verlag, LNCS 1416, 1998.
- [BOU 99a] BOURJOT C., CHEVRIER V., BERNARD A., KRAFFT B., « Coordination par le biais de l'environnement: une approche biologique », dans Gleizes M.P. et Marcenac P. (dir.), *Ingénierie des systèmes multi-agents. JFIADSMASMA'99*, Hermès, Paris, p. 237-250, 1999.

- [BOU 99b] BOUSQUET F., BARRETEAU O. *et al.*, « An environmental modelling approach. The use of multi-agents simulations », dans F. Blasco et A. Weill (dir.), *Advances in Environmental and Ecological Modelling*, Elsevier, p. 113-122, 1999.
- [BOU 01] BOUSQUET F., LE PAGE C., BAKAM I., TAKFORYAN A., « Multi-agent simulations of hunting wild meat in a village in eastern Cameroon », *Ecological modelling*, n° 138, p. 331-346, 2001.
- [BRO 91] BROOKS R., « Intelligence without representation », *Artificial Intelligence*, n° 47, p. 139-159, 1991.
- [BRO 94] BROWN J., « Complex ecological systems », dans G. Cowan, D. Pines (dir.), *Complexity: metaphrs models and reality*, Addison-Wesley, n° XIX, p. 419-443, 1994.
- [BRU 92] BRUNET R., FERRAS R., THERY H., *Les mots de la géographie*, Reclus-La Documentation Française, Montpellier-Paris, 1992.
- [CAR 96] CARTER J., MOAB: A generalizable artificial intelligence model for simulating duck nest depredation in the northern paririe region of North America, Massachusetts, 390, 1996.
- [CAZ 00] CAZOULAT R., VICTORRI B., Etude de la dynamique des populations par simulation, <http://www.univ-caen.fr>, 2000.
- [COL 92] COLLINS R.J., JEFFERSON D.R., « AntFarm: towards simulated evolution », dans *Artificial Life II*, C. Langton, C. Taylor, J. Doyne Farmer et S. Rasmussen, Addison-Wesley, p. 579-602, 1992.
- [COR 94] CORBARA B., « Des fourmis aux agents aux robots : pour une sociéthologie générale », dans Pavard (dir.), *Systèmes coopératifs : de la modélisation à la conception*, Octarès, Toulouse, p. 227-241, 1994.
- [COS 93] COSTANZA R., WAINGER L. *et al.*, « Modelling complex ecological economic systems: toward an evolutionary, dynamic understanding of humans and nature », *BioScience*, 1993.
- [DAG 00] DAGORN L., MENCZER F. *et al.*, « Co-evolution of movement behaviours by tropical pelagic predatory fishes in response to prey environment: a simulation model », *Ecological modelling*, vol. 134, n° 2-3, p. 325-341, 2000.
- [DEA 00] DEAN J., GUMERMAN G. *et al.*, « Understanding Anasazi culture change through Agent-Based Modeling », dans T. Kohler et G. Gumerman (dir.), *Dynamics in Human and Primate societies*, Oxford University Press, 2000.
- [DEA 79] DEANGELIS D.L., COX D.C. *et al.*, « Cannibalism and size dispersal in the young-of-the year largemouth bass: experiments and model », *Ecologica Modelling*, n° 8, p. 133-148, 1979.
- [DEA 92] DEANGELIS D.L., GROSS L.J., *Individual-based models and approaches in ecology*, Chapman and Hall, 1992.
- [DEN 89] DENEUBOURG J.L., GOSS S., « Collective pattrens and decision making », *Ethol. Ecol. & Evol*, n° 1, p. 295-311, 1989.
- [DER 98] DERRY, « Modelling ecological interaction despite object-oriented modularity », *Ecological modelling*, n° 107, p. 145-158, 1998.

- [DEU 97] DEUTSCHMAN D.H., LEVIN S., DEVINE C., BUTTEL L., Scaling from trees to forests : analysis of a complex simulation model, 1997, (<http://www.sciencemag.org/feature/data/deutschman/abstract>).
- [DOR 93] DORAN J., PALMER M., « The {EOS} Project : Integrating Two Models of Paleolithic Social Change », dans N. Gilbert et R. Conte (dir.), *Artificial Societies*, UCL Press, 1993.
- [DRO 93] DROGOU A., De la simulation multi-agent à la résolution collective de problèmes. Une étude de l'émergence de structures d'organisation dans les systèmes multi-agent, Thèse de doctorat, Université Paris VI, 1993.
- [DRO 94] DROGOU A., « La Mémoire Collective Dans Les Sma Réactifs : L'exemple de Manta », *Mémoire Collective, Dialogue Entre Les Sma Artificiels et Les Sciences Humaines et Biologiques*, Centre de recherche de Royallieu, Compiègne, 1994.
- [DUM 01] DUMONT B., HILL D., « Multi-agent simulation of group foraging in sheep: effects of spatial memory, conspecific attraction and plot size », *Ecological modelling*, n° 141, p. 201-215, 2001.
- [DUR 94] DURETT R., LEVIN S., « The importance of being discrete (and spatial) », *Theoretical population biology*, n° 46, p. 363-394, 1994.
- [EPS 96] EPSTEIN J., AXTELL R., *Growing Artificial Societies. Social Science from the Bottom Up*, Brookins Institution Press/The MIT Press, 1996.
- [FER 98] FERRAND N., *SMAGET, Systèmes multi-agents gestion du territoire*, Cemagref Editions, 1998.
- [FER 97] FERRAND N., DEMAZEY Y. *et al.*, « Systèmes multi-agents réactifs et résolution de problèmes spatialisés », *Revue d'Intelligence Artificielle*, vol. 12, n° 1, p. 37-72, 1997.
- [FOL 89] FOLSE L., PACKARD J., GRANT W., « AI modelling of animal movements in a heterogenous habitat », *Ecological modelling*, n° 46, p. 57-72, 1989.
- [FRA 95] FRANCHESQUIN N., Modélisation des négociations aboutissant à une prise de décision collective au sein d'une assemblée communautaire, DEA de l'université Paris 6, 1995.
- [FRO 93] FRONTIER S., PICHOD-VIALE D., *Ecosystèmes, structure fonctionnement évolution*, Masson, 1993.
- [FRY 97] FRYXELL J.M., LUNDBERG P., *Individual behavior and community dynamics*, Chapman et Hall, Londres, 1997.
- [GAR 70] GARDNER M.R., ASHBY W.R., « Connectance of large dynamic (cybernetic) systems: critical values for stability », *Nature*, n° 228 (784), 1970.
- [GIL 95] GILBERT N., « Emergence in social simulation », dans N. Gilbert et R. Conte (dir.) *Artificial societies. The computer simulation of social life*, UCL Press, p. 144-156, 1995.
- [GIM 98] GIMBLETT H., RICHARDS M., ITAMI R., « A complex systems approach to simulating human behaviour using synthtic landscapes », *Complexity international*, n° 6, 1998.
- [GIN 98] GINOT V., LE PAGE C., « Mobydic, a generic multi-agents simulator for modelling populations dynamics », dans A. Pasqual del Pobil, J. Mira et A. Moonis (dir.), *Tasks and methods in applied artificial intelligence*, Springer-Verlag, LNCS 1416, 1998.

- [GRA 98] GRAND S., CLIFF D., « Creatures: entertainment software agents with artificial life », *Autonomous agents and multi-agent systems*, n° 1, p. 39-57, 1998.
- [GRI 99a] GRIMM V., « Ten years of individual-based modelling in ecology: what we have learned and what could we learn in the future? », *Ecological modelling*, n° 115, p. 129-148, 1999.
- [GRI 99b] GRIMM V., WYSZOMIRSKI T. *et al.*, « Individual-based modelling and ecological theory: synthesis of a workshop », *Ecological modelling*, n° 115, p. 275-282, 1999.
- [GRO 98a] GRONEWOLD A., SONNENSCHNEIN M., « Event-based modelling of ecological systems with asynchronous cellular automata », *Ecological Modelling*, n° 108, p. 37-52, 1998.
- [GRO 98b] GROSS, Behavioral ecology and individual-based modelling, Lecture Notes for Beijer Institute, 1998 (<http://www.tiem.utk.edu/~gross/behav.txt>).
- [GUE 99] GUERRIN F., COURDIER R. *et al.*, « Biomas : un modèle multi-agents pour aider à la gestion négociée d'effluents d'élevage », dans N. Ferrand (dir.), *Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires*, Cemagref éditions, 1999.
- [GUI 99] GUILLOT A., « Pour une approche dynamique des animats », dans A. Drogoul et J.A. Meyer (dir.), *Intelligence artificielle située*, Hermès, Paris, 1999.
- [GUT 97] GUTKNECHT O., FERBER J., {MadKit}: Organizing heterogeneity with groups in a platform for multiple multi-agent systems, Rapport Interne, LIRMM, 1997.
- [HAR 68] HARDIN G., « The tragedy of the commons », *Science*, n° 162, p. 1243-1248, 1968.
- [HIL 95] HILL D., « Verification & validation of ecosystem simulation models », *Proceedings of the SCS'95*, Ottawa, Ontario, 1995.
- [HOG 88] HOGEWEG P., « Mirror beyond mirror: puddles of life », dans C. Langton (dir.), *Artificial life*, Addison-Wesley, 1988.
- [HOG 90] HOGEWEG P., HESPER B., « Individual-oriented modelling in ecology », *Mathl. Comput. Modelling*, n° 13, p. 83-90, 1990.
- [HOL 73] HOLLING, « Resilience and stability of ecological systems », *Annual Review of Ecological Systems*, n° 4, p. 1-23, 1973.
- [HOL 75] HOLLAND J., *Adaptation in natural and artificial systems*, University of Michigan Press, 1975.
- [HOL 87] HOLLING, « Simplifying the complex: the paradigms of ecological function and structure », *European Journal of Operational Research*, n° 30, p. 139-146, 1987.
- [HRA 97] HRABER P., JONES T., FOREST S., « The ecology of Echo », *Artificial life*, n° 3, p. 165-190, 1997.
- [HUS 88] HUSTON M., DEANGELIS D. *et al.*, « New computer models unify ecological theory », *Bioscience*, n° 38, p. 682-691, 1998.
- [HUT 00] HUTZLER G., Du jardin des hasards au jardin des données : une approche artistique et multi-agent des interfaces homme/systèmes complexes, Thèse de doctorat, Université Paris 6, 2000.

- [JAG 00] JAGER W., *Modelling consumer behaviour*, Universal Press, 2000.
- [JAN 00] JANSSEN M., WALKER B. *et al.*, « An adaptive agent model for analysing co-evolution of management and policies in a complex rangeland system », *Ecological Modelling*, n° 131, p. 249-268, 2000.
- [JEFF 91] JEFFERSON D., COLLINS R. *et al.*, « Evolution as a theme in Artificial life: the genesys/tracker system », dans C. Lanton, C. Taylor, J. Farmer et S. Rasmussen (dir.), *Artificial life II*, Addison-Wesley, n° X, p. 549-578, 1991.
- [JUD 94] JUDSON O.P., « The rise of the individual-based model in ecology », *TREE*, n° 9, p. 9-14, 1994.
- [KAW 94] KAWATA M., TOQUENAGA Y., « From artificial individuals to global patterns », *TREE*, n° 9, 1994.
- [KEI 97] KEITT T., « Stability and complexity on a lattice: coexistence of species in an individual-based food web model », *Ecological modelling*, n° 102, p. 243-258, 1997.
- [KEY 39] KEYNES J.M., *Théorie générale de l'emploi, de l'intérêt et de la monnaie*, Petite Bibliothèque Payot, 1939.
- [KOH 96] KOHLER T.A., CARR E., « Swarm based modelling of prehistoric sttlement systems in southwestern North America », *Archaeological applications of GIS, UISPP XIIIth Congress*, Forli, Italie, 1996.
- [KOH 00a] KOHLER T., KRESL J. *et al.*, « Be there and then: a modeling approach to settlement determinants and spatial efficiency among late ancestral pueblo populations of the Mesa Verde region, US SouthWest », dans T. Kohler et G. Gumerman (dir.), *Dynamics in human and primate societies*, Oxford University Press, 2000.
- [KOH 00b] KOHLER T.A., GUMERMAN G.J. (dir.), *Dynamics in Human and Primate societies. Agent-based modeling of social and spatial processes*, Santa Fe Institute studies in the sciences of complexity, Oxford University Press, 2000.
- [KOZ 94] KOZA J., « Artificial life: spontaneous emergence of self-replicating and evolutionary self improving computer programs », dans C. Langton (dir.), *Artificial life III*, Addison-Wesley, 1994.
- [KOZ 99] KOZLAK J., DEMAZEAU Y. *et al.*, « Multi-Agent system to model the Fishbank play process », *Proceedings of CEEMAS'99*, SPIIRAS, Saint-Petersbourg, p. 154-162, 1999.
- [KRE 96] KREBS F., BOSSEL H., « Emergent value orientation in self-organization of an animat », *Ecological modelling*, n° 96, p. 143-164, 1994.
- [KRE 98] KREFT J.U., BOOTH G. *et al.*, BacSim, a simulator for individual-based modelling of bacterial colony growth, 1998 (<http://www.eeb.yale.edu/ginger/bacillus/node1.html>).
- [LAM 99] LAMBERT P., ROCHARD E. *et al.*, « Environnement des poissons migrateurs dans les modèles individus-centrés », dans N. Ferrand (dir.), *Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires*, Cemagref éditions, 1999.
- [LAN 88] LANGTON C., « Artificial Life », dans C. Langton (dir.), *Artificial life*, Addison-Wesley, 1998.

- [LEB 99] LE BER F., DURY A. *et al.*, « Simuler l'organisation spatiale d'un territoire agricoles : différentes approches », dans N. Ferrand (dir.), *Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires*, Cemagref éditions, 1999.
- [LEM 89] LE MOIGNE J.L., *La modélisation des systèmes complexes*, Dunod, 1989.
- [LEP 96] LE PAGE C., CURY P., «How spatial heterogeneity influences population dynamics: simulations in SeaLab », *Adaptive behavior*, vol. 4, n° 3/4, p. 255-281, 1996.
- [LEP 99] LE PAGE C., LARDON S. *et al.*, « Entités spatiales génériques et modèles de simulation multi-agent », dans *Ingénierie des systèmes multi-agents JFIADSM'99*, Hermès, p. 341-342, 1999.
- [LET 99] LETT C., SILBER C. *et al.*, « Comparison of a cellular automata network and an individual-based model for the simulation of forest dynamics », *Ecological modelling*, n° 121, p. 277-293, 1999.
- [LIN 94] LINDGREN K., NORDAHL M., « Artificial food webs », dans C. Langton (dir.), *Artificial life III*, Addison Wesley, 1994.
- [LIU 98] LIU J., ASHTON P., « FORMOSAIC: an individual-based spatially explicit model for simulating forest dynamics in landscape mosaics », *Ecological modelling*, n° 106, p. 177-200, 1989.
- [LOM 99] LOMNICKI A., « Individual-based models and the individual-based approach to population ecology », *Ecological modelling*, n° 115, p. 191-198, 1999.
- [LOR 99] LOREK H., SONNENSCHIN M., « Modelling and simulation software to support individual-based ecological modelling », *Ecological modelling*, n° 115, p. 199-216, 1999.
- [MAE 97] MAES P., SCHNEIDERMAN B., « Direct Manipulation vs. Interface Agents: a Debate », *Interactions*, vol. 4, n° 6, 1997.
- [MAL 93] MALEY C., CASWELL H., « Implementing i-state configuration models for population dynamics: an object-oriented programming approach », *Ecological modelling*, n° 68, p. 75-89, 1993.
- [MAR 97] MARCENAC P., « Modélisation de systèmes complexes par agents », *TSI*, vol. 16, n° 8, p. 1013-1038, 1997.
- [MAT 00] MATHEVET R., Usages des zones humides camarguaises : enjeux et dynamique des interactions Environnement/Usagers/Territoire. Lyon, Thèse de l'université Jean Moulin Lyon 3, 2000.
- [MAY 73] MAY R., *Stability and complexity in model ecosystems*, Princeton University Press, 1973.
- [MCC 93] MCCAULEY E., WILSON W.G. *et al.*, « Dynamics of age-structured and spatially structured predator-prey interactions: individual-based models and population-level formulation », *The american naturalist*, n° 142, p. 412-442, 1993.
- [MEC 98] MECHOUD S., HILL D. *et al.*, « Simulation multi-agents de l'entretien des paysages par des herbivores en moyenne montagne », dans N. Ferrand (dir.), *Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires*, Cemagref éditions, 1998.
- [MEN 95] MENARD C., *L'Economie des organisations*, Editions La Découverte, 1995.

- [MIN 96] MINAR N., BURKHART R. *et al.*, The swarm simulation system: a toolkit for building multi-agent simulations, 1996 (<http://www.santafe.edu/project/swarm>).
- [MOR 77] MORIN E., *La méthode, la nature de la nature*, Paris, Le Seuil, 1977.
- [NIS 97] NISHIMURA S.I., TAKASHI IkeGAMI., « Emergence of collective strategies in a prey-predator game model », *Artificial life*, n° 3, p. 243-260, 1977.
- [ODU 57] ODUM H.T., « Trophic structure and productivity of Silver Springs Florida », *Ecol. Monogr*, n° 27, p. 55-112, 1957.
- [OPP 88] OPPENHEIMER P., « The artificial menagerie », dans C. Langton (dir.), *Artificial Life*, Addison-Wesley, p. 251-275, 1988.
- [PEP 99] PEPPER J., SMUTS B., « The evolution of cooperation in an ecological context : an agent based-model », dans T. Kohler et G. Gunnerman (dir.), *Dynamics of human and primate societies: agent based-modelling of social and spatial processes*, New York, Oxford University Press, 1999.
- [PER 97] PERRIER E., CAMBIER C., « Une approche multi-agents pour simuler les interactions entre des acteurs hétérogènes de l'infiltration et du ruissellement d'eau sur une surface de sol », dans F. Blasco (dir.), *Tendances nouvelles en modélisation pour l'environnement*, Elsevier, p. 233-242, 1997.
- [PIG 20] PIGOU, *The economics of welfare*, Mc Millan, 1920.
- [RAM 98] RAMAT E. PREUX P., *et al.*, « Modélisation multi-agents de systèmes naturels – réflexions générale et application en biologie marine », dans N. Ferrand (dir.), *Modèles et systèmes multi-agents pour la gestion de l'environnement et des territoires*, Cemagref éditions, 1998.
- [REY 87] REYNOLDS C., « Flocks, herds and schools: a distributed behavioral model », *Computer graphics*, vol. 21, n° 4, p. 25-34, 1987.
- [RIS 00] RISENHOOVER K., Deer Management Simulator, <http://lutra.tamu.edu/dms/dms.html>, 2001.
- [ROE 91] ROESE H., RISENHOOVER K. *et al.*, « Habitat heterogeneity and foraging efficiency: an individual-based model », *Ecological modelling*, n° 57, p. 133-143, 1991.
- [ROU 01] ROUCHIER J., BOUSQUET F. *et al.*, « A multi-agent model for transhumance in North Cameroon », *Journal of Economic Dynamics and Control*, n° 25, p. 527-559, 2001.
- [SAA 88] SAARENMA H., DTONE N. *et al.*, « Artificial intelligence approaches to modelling animal/habitat interactions », *Ecological modelling*, n° 44, p. 125-141, 1988.
- [SAV 00] SAVAGE M., BELL R., Arborgames, <http://margay.sscnet.ucla.edu/~reb/arborweb/arbormain.html>, 2000.
- [SCH 97] SCHMITZ O.J., BOOTH G., « Modelling food web complexity: the consequences of individual-based, spatially explicit behavioural ecology on trophic interactions », *Evolutionary Ecology*, n° 11, p. 379-398, 1997.
- [SHI 00] SHIN Y., Interactions trophiques et dynamiques des populations dans les écosystèmes marins exploités. Approche par modélisation individus-centrée, Thèse de doctorat, Université Paris 7, 2000.

- [SIL 93] SILVERT W., « Object-Oriented ecosystem modelling », *Ecological modelling*, n° 68, p. 91-118, 1993.
- [THE 94] THÉRAULAZ G., « Du super organisme à l'intelligence en essaim : modèles et représentations du fonctionnement des sociétés d'insectes », dans E. Bonabeau et G. Theraulaz (dir.), *Intelligence collective*, Paris, Hermès, p. 29-109, 1994.
- [TYR 93] TYRELL T., Computational mechanism for action selection, University of Edimburgh, 1993.
- [UCH 96] UCHMANSKI J., GRIMM V., « Individual-based modelling in ecology: what makes the difference? », *TREE*, n° 11, p. 437-441, 1996.
- [VBE 68] VON BERTALLANFFY L., *Théorie générale des systèmes*, Editions Dunod, Paris, 1968.
- [VIL 92] VILLA F., « New computer Architectures as tools for ecological thought », *TREE*, n° 7, p. 179-183, 1992.
- [VPA 98] VAN DYKE PARUNAK H., SAVIT R. *et al.*, « Agent based modelling vs equation based modelling: a case study and user's guide », dans J. Sichman, R. Conte et N. Gilbert (dir.), *Multi agent systems and agent based simulation*, Springer-Verlag, n° 1534, 1998.
- [WIE 91a] WEISBUCH G., *Complex systems dynamics*, Perseus book, 1991.
- [WIE 91b] WEISBUCH G., « Systèmes complexes et comportement générique », dans F. Fogelman Soulié (dir.), *Les théories de la complexité. Autour de l'œuvre de Henri Atlan*, Seuil, 1991.
- [WEI 97] WEISBUCH G., KIRMAN A. *et al.*, *Market Organization*, Lecture notes in economics and mathematical systems, n° 456, 1997.
- [WEL 89] WELCOMME R., *Fisheries ecology of foodplain rivers*, Longman, Londres, 1989.
- [WEL 96] WELLMAN M., « Market-oriented programming: Some early lessons », *Market-Based Control: A Paradigm for Distributed Resource Allocation*, S. Clearwater, World Scientific, 1996.
- [WIL 93] WILSON W., ROOS D. *et al.*, « Spatial instabilities within the diffusive lotka-Volterra system: individual-based simulation results », *Theoretical population biology*, n° 43, p. 91-127, 1993.
- [WIL 96] WILSON W.G., « Lotka's game in predator-prey theory: linking populations to individuals », *Theoretical population biology*, n° 50, p. 368-393, 1996.

Index

A

ACL 114, 119
acte de discours (de langage) 49, 50, 86,
94, 95, 114, 115, 212, 217, 225, 228,
231
action 29, 39, 51, 71, 76, 79, 80, 83, 85,
89, 139
agent 18, 27, 71, 73, 74, 177
assistant 20, 24
autonome 77, 79
délibératif 78, 81, 86, 91
hybride 78, 83, 88
interagissant 77, 84
mobile 24
rationnel 20
réactif 78, 79, 85, 90
social 77, 90
agentification 21
apprentissage 24, 171
architecture 25, 30, 31, 49, 55, 60, 73, 84,
144, 177, 186
autonome, autonomie 17, 20, 24, 27, 32,
72, 154
auto-organisation 201

B

BD1 53, 73, 74, 81, 82, 83, 87, 89, 92
bibliothèque 181, 196, 197
but 33, 46, 83, 91, 151

C

communication 25, 39, 45, 46, 47, 49, 52,
85, 89, 109, 111, 144, 215, 217, 222,
224, 226, 228
compétence 87, 189
complexité 141, 177, 231
composant 185, 191, 197, 201
conception 52, 53
connaissance 28, 32, 36, 55, 73, 139
contrôle 72, 80, 84, 88, 89, 93, 95, 97,
141, 179, 181, 188, 189, 194
conversation 76
coordination 25, 40, 41, 45, 72, 75, 76,
95, 139, 248

D

décision 139
décomposition 97
délibération 78, 144
description 72, 74
développement 25, 52, 53, 178, 179, 180,
189, 190, 192
distribution 151, 179, 188, 189

E

écologie 236, 237, 238, 239, 240, 241,
242, 243, 244, 245, 246, 247, 249,
250, 253, 255, 256, 258
engagement 73, 87, 94, 95

environnement 18, 28, 45, 71, 76, 79, 88, 236, 237, 240, 241, 244, 245, 246, 247, 249, 250, 252, 253, 254, 255, 256, 257, 258
de développement 55, 74
exécution 189, 192, 193, 199, 200

F, H

FIPA 114
hétérogénéité 178, 188, 201

I

individu 236, 238, 239, 240, 243, 255, 256, 258
interaction 25, 30, 33, 37, 39, 40, 76, 85, 94, 110
Internet 207, 210, 212, 217, 219, 221, 228, 232
interopérabilité 209, 210, 211, 212, 217, 227, 228

K, L

KQML 117
langage
de programmation, 50, 52, 74, 141, 177, 180, 181, 182, 186
de communication (d'interaction) 45, 76, 114, 191, 228

M, N

modèle (modélisation) 29, 36, 39, 43, 46, 51, 53, 54, 72, 140, 146
modularité, module 188, 189
négociation 40, 44, 46

O

objet 18, 177, 178, 180, 181, 183, 185, 193
ontologie 213, 215, 224, 228, 232
organisation 17, 24, 76, 77, 90, 95, 187, 188, 197, 235, 236, 238, 239, 243, 245, 249, 250, 251, 252, 253, 256, 258

P

perception 71, 76, 79, 80, 83, 85, 89, 140
performatif 116, 119
personnalisation 210, 217, 225, 232
plan (planification) 29, 39, 45, 46, 76, 83, 84, 87, 88, 89, 140
programmation 34, 180, 181, 183, 184, 186, 198
protocole
de communication 36, 44, 46, 155, 209, 212, 213, 215, 219, 227, 230
d'interaction 76, 86, 87, 88, 89, 95, 124, 128

R

rationalité 156
réaction 140
règle 181, 187
représentation 72, 144
ressource 241, 244, 246, 248, 249, 250, 252, 253, 254, 255, 256

S

service 54, 56
simulation 25, 79, 239, 240, 242, 243, 244, 245, 249, 252, 253, 254, 257
social 236, 237, 249, 250, 251, 252, 257
spatial 240, 243, 248, 250, 253, 255, 256
standard 209, 211, 215, 220, 227, 228
subsumption 188
supervision 209, 225
synchronisation 144

T

tableau noir 186
tâche 79, 88, 94, 95, 143
télécommunication 25, 207, 208, 209, 210, 218, 227