

N° d'ordre : 2890

THÈSE

présentée devant

l'Université Paul Sabatier de Toulouse (sciences)

en vue de l'obtention du

Doctorat de l'Université Paul Sabatier

Spécialité : Informatique

par

Valérie CAMPS

**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**

Soutenue le lundi 5 janvier 1998 devant la commission d'examen composée de :

M. VIGNOLLE J.	Président
M. PAVARD B.	Directeur de thèse
M. HATON J.P.	Rapporteur
Mme SAYETTAT Cl.	Rapporteur
M. DEMAZEAU Y.	
Mme GLEIZES M.P.	
M. GLIZE P.	
M. LEGER A.	

Laboratoire : Institut de Recherche en Informatique de Toulouse

Université Paul Sabatier, 118 route de Narbonne, 31062 TOULOUSE CEDEX

Résumé : Les techniques multi-agents permettent de construire des systèmes artificiels dans lesquels l'activité collective observée n'est décrite dans aucune de leurs parties. Cette approche est particulièrement utile dans les applications où la coopération d'une multitude de tâches élémentaires est indispensable pour la réalisation d'une activité collective précise mais difficilement formalisable, lorsque le système est fortement évolutif ou encore lorsque l'environnement exerce des contraintes fortes et dynamiques sur le système.

Pour aboutir à un comportement collectif cohérent alors que les agents ne cherchent à atteindre qu'un objectif individuel, la propriété suivante a été démontrée : "pour tout système fonctionnellement adéquat, il est possible de trouver au moins un système à milieu intérieur coopératif qui lui soit équivalent", c'est-à-dire un système dont toutes les parties entretiennent des relations coopératives entre elles et avec l'environnement du système.

Cette propriété conduit à une nouvelle méthode de conception de systèmes. En effet, il ne devient plus nécessaire de connaître la finalité d'un système mais de savoir que tous les éléments qui le constituent doivent coopérer. Dès qu'une situation non coopérative survient, le système doit s'auto-organiser afin de revenir à un état coopératif. La tâche du concepteur d'une application consiste alors à identifier les situations non coopératives auxquelles peuvent être confrontés les éléments du système puis à définir les traitements qu'ils doivent effectuer pour revenir à un état coopératif conformément à la théorie présentée.

Ces principes ont été appliqués au projet "ARCADIA", dans le cadre d'un contrat CNET réalisé en collaboration avec l'ONERA-CERT. Son but est de permettre d'accéder à de l'information pertinente dans un réseau de ressources d'informations distribuées, hétérogènes et en constante évolution. Des exemples de fonctionnement du prototype réalisé sont analysés.

Mots clés : intelligence artificielle distribuée, systèmes multi-agents, auto-organisation, apprentissage, coopération, recherche d'information.

Abstract : Multi-agent methods allow to build artificial systems where collective activity is observed in any of their parts. This approach is especially useful in applications where cooperation of a multitude of elementary tasks is indispensable for the realization of a global collective activity formalizable with difficulty, when the system is strongly evolutionary or when the environment brings strong and dynamic constraints to bear on the system.

In order to lead to a coherent collective behavior whereas agents only seek to reach an individual objective, the following property has been proved : when every element making up a system keeps cooperative relations with each other, the system meets the requirements for which it was designed (the system is functionally adequate). This observation has induced to elaborate and prove the following theorem : “to any functionally adequate system in a given environment can be associated at least one system having a cooperative internal medium and achieving an equivalent function”, namely a system from which all parts have cooperative relations between themselves and the environment of the system.

This method can be interpreted as a new way of conceiving a system. Thus, knowing the global function of a system becomes secondary, the point being to know all of its parts must cooperate one with the other. As soon as an uncooperative situation occurs, the system must self-organize by itself in order to resume with a cooperative state. The task of the designer then consists in identifying and instanciating uncooperative situations which may occur, and in defining the operations each element has to process when being in such situations.

These principles have been applied to a project called “ARCADIA” in the framework of a contract CNET realized in collaboration with the ONERA-CERT. The purpose of this project is to allow the retrieval of relevant information from a distributed information resources network, where resources are heterogeneous by their content and permanently changing. Significant examples of the realized prototype are analyzed.

Key words : distributed artificial intelligence, multi-agent systems, self-organization, learning, cooperation, information retrieval.

Je tiens à remercier chaleureusement toutes les personnes que j'ai eu à côtoyer tout au long de ces années de recherche pour leur aide à la fois technique, psychologique et matérielle.

Je remercie, Monsieur Bernard Pavard, directeur de recherche CNRS pour avoir été mon directeur de recherche et Monsieur Jean Vignolle, professeur à l'Université Paul Sabatier et directeur de l'IRIT, pour m'avoir accueillie dans son laboratoire et pour avoir accepté de présider le jury de ma thèse.

J'exprime ma reconnaissance à :

Yves Demazeau, chargé de recherche CNRS à l'Institut IMAG à Grenoble,

Marie-Pierre Gleizes, maître de conférences à l'Université Paul Sabatier,

Pierre Glize, ingénieur CNRS à l'IRIT,

Jean-Paul Haton, professeur à l'Université de Nancy 1,

Alain Léger, ingénieur CNET au CCETT de Rennes,

Claudette Sayettat, professeur à l'Ecole des Mines de Saint Etienne

pour l'honneur qu'ils m'ont fait en participant au jury de cette thèse.

Je tiens tout particulièrement à remercier Claudette Sayettat et Jean Paul Haton qui ont bien voulu s'intéresser à ce travail en acceptant d'être rapporteurs de ma thèse ainsi que pour leur lecture minutieuse et leurs remarques pertinentes. Un grand merci également à Yves Demazeau pour l'attention avec laquelle il a relu mon manuscrit et à Alain Léger, pour avoir prêté tant d'intérêt au projet ARCADIA, à son aide et à ses remarques pertinentes.

J'exprime toute ma gratitude à Pierre Glize qui a encadré cette thèse et le DEA qui a précédé. Je le remercie pour m'avoir accueillie au sein de son équipe et m'avoir accordé sa confiance. Il m'a tout fait découvrir d'un domaine de recherche passionnant et m'a permis, à l'aide de précieux conseils de mener à bien ce travail.

Un grand merci à André Machonin, sans qui ARCADIA se serait pas ce qu'elle est. Il m'a tout appris de la programmation objet et m'a ô combien de fois aidée à trouver pourquoi mes petits agents ne voulaient pas coopérer avec moi !!! Sa disponibilité a été sans faille.

Je remercie Marie-Pierre Gleizes pour ses remarques constructives, l'apport de son expérience et pour la disponibilité et l'aide qu'elle m'a apporté dans ma charge d'enseignement.

Je remercie également Bernard Carpuat et Christine Régis à la fois pour la relecture de ce manuscrit, pour leur soutien, leurs encouragements et leurs conseils.

Je n'oublie pas les autres membres de l'équipe Systèmes Multi-Agent Coopératifs pour l'atmosphère de travail agréable dont j'ai bénéficiée. Un merci tout particulier à Jean-Luc Soubie pour tous ses conseils et sa disponibilité dans mes moments de panique.

Un grand merci également à tous les participants au projet ARCADIA pour leur aide, leurs remarques et précisions. Un merci tout particulier à Pierre Bieber et Patrice Cros pour leurs encouragements, leurs contributions ainsi que les discussions et travaux arcadiens réalisés toujours dans la bonne humeur.

Toute mon affection à Christine Piquemal-Baluard, mon ex-compagne de bureau. Malgré la distance qui nous sépare maintenant, elle a su être présente pour m'encourager, me conseiller, et me divertir. Un grand clin d'oeil à Sylvie Trouilhet, pour sa bonne humeur et sa gentillesse qui m'ont à maintes reprises bien aidée. Un grand merci pour avoir relu mon manuscrit avec tant d'attention.

Merci également à Pascale N'Guyen pour sa gentillesse et sa disponibilité, ainsi qu'à Jean-Pierre Ceccato et Jean-Pierre Baritaud pour leur efficacité et leur diligence. Qu'ils m'excusent de les avoir parfois trop "bousculés".

Enfin, de manière bien plus personnelle je remercie tous les amis qui m'ont aidée de près ou de loin durant ces années. J'espère qu'ils se reconnaîtront même si je ne les cite pas et qu'ils m'excuseront de ne pas avoir été très présente ces derniers temps.

Un grand merci à ma famille et tout particulièrement à mes parents pour m'avoir permis de faire des études longues dans les meilleures conditions qu'il soit ainsi que pour m'avoir encouragée et soutenue pendant toutes ces années.

Enfin, un grand merci à mon "chef de gare" du dimanche qui a su supporter mon humeur bien souvent abominable ces derniers temps et un rythme de vie plus que sédentaire et studieux ces derniers mois. Merci d'avoir contribué à la réussite de ce travail, d'y avoir accordé tant d'attention et pour tout le soutien prodigué dans les moments les plus difficiles.

TABLE DES MATIÈRES

INTRODUCTION

1. Cadre de l'étude	1
2. La problématique étudiée	2
3. Les principales définitions employées.....	4
4. Présentation de la thèse	6

PARTIE 1 9

Chapitre 1 : La coopération dans les systèmes multi-agents

1.1. Introduction.....	9
1.2. Cadre de l'étude.....	10
1.2.1. La coopération.....	10
1.2.2. La coordination.....	11
1.2.3. La collaboration	13
1.3. Étude des différents travaux	13
1.3.1. L'apport de la coopération pour la résolution.....	14
1.3.1.1. Les travaux de Bouron & Ferber.....	14
1.3.1.2. La méthode Goldman & Rosenschein	14
1.3.1.3. La méthode de Prasad, Lander, Lesser	15
1.3.1.4. Les travaux de Hogg & Huberman et Ebstyne	17
1.3.2. La coopération en tant qu'attitude sociale	18
1.3.2.1. Les travaux de Cesta, Miceli, Rizzo.....	18
1.3.2.2. Les travaux de Kalenka & Jennings.....	20
1.3.2.3. Les travaux de Sekaran & Sen	21
1.4. Discussion.....	22

Chapitre 2 : L'apprentissage dans les systèmes multi-agents

2.1. Introduction.....	25
2.2. Cadre de l'étude.....	26
2.3. L'apprentissage par renforcement.....	28
2.3.1. L'algorithme du Q-learning	29
2.3.2. Les systèmes classifieurs.....	30
2.3.3. L'algorithme du "bucket brigade"	31

2.4. L'apprentissage des connaissances du domaine.....	31
2.4.1. Méthodes d'apprentissage sur les compétences.....	32
2.4.1.1. Le modèle de Sekaran & Sen.....	33
2.4.1.2. Le système de Mataric.....	33
2.4.1.3. Le modèle de Sian	35
2.4.2. La distribution des algorithmes	36
2.4.3. L'asynchronisme des algorithmes	36
2.4.4. La coopération dans les algorithmes	37
2.5. L'apprentissage de l'organisation	38
2.5.1. Méthodes d'apprentissage sur l'organisation.....	38
2.5.1.1. La méthode de Shaw & Whinston	38
2.5.1.2. La méthode de Weiß	39
2.5.1.3. La méthode de Foisel, Chevrier, Haton.....	41
2.5.1.4. Le modèle de Prasad, Lander, Lesser	42
2.5.1.5. Le modèle de Terabe, Washio, Katai, Sawaragi.....	43
2.5.2. La distribution des algorithmes.....	44
2.5.3. L'asynchronisme des algorithmes d'apprentissage proposés	45
2.5.4. La coopération dans les algorithmes	45
2.6. Discussion	46

Chapitre 3 : L'auto-organisation : état de l'art et principe

3.1. Introduction.....	49
3.2. Quelques définitions	51
3.3. Notions fondamentales	52
3.3.1. Caractéristiques essentielles d'un système.....	53
3.3.2. Organisation et auto-organisation d'un système	56
3.3.3. Caractéristiques des systèmes auto-organiseurs.....	57
3.4. L'auto-organisation dans les systèmes naturels.....	58
3.4.1. Auto-organisation par auto-catalyse	59
3.4.2. Auto-organisation par stigmergie	60
3.5. L'auto-organisation dans le domaine de la systémique	62
3.5.1. L'autopoïèse et l'auto-organisation.....	62
3.5.2. L'homéostasie et l'auto-organisation.....	62
3.6. L'auto-organisation dans les modèles connexionnistes	63
3.6.1. Les réseaux adaptatifs non linéaires.....	64
3.6.2. Les cartes auto-organisatrices	65

3.7. L'auto-organisation dans les algorithmes génétiques	66
3.7.1. L'opérateur de sélection	67
3.7.2. L'opérateur de croisement.....	67
3.7.3. L'opérateur de mutation	67
3.8. L'auto-organisation en robotique	68
3.8.1. Auto-organisation par auto-catalyse.....	68
3.8.2. Auto-organisation spatiale par attraction/répulsion	69
3.9. L'auto-organisation en Intelligence Artificielle	70
3.9.1. Auto-organisation par décomposition/recomposition.....	70
3.9.2. Auto-organisation par systèmes multi-agents récursifs.....	72
3.9.3. Auto-organisation par rétroaction.....	73
3.9.4. Auto-organisation par induction	74
3.10. Émergence	75
3.11. Discussion	77

PARTIE 2

79

Chapitre 4 : Modèle d'apprentissage dans un système multi-agent coopératif

4.1. Introduction	79
4.2. Cadre d'étude	80
4.2.1. Propriété à démontrer	80
4.2.2. Hypothèses	81
4.2.3. Contexte.....	81
4.2.4. Notations.....	83
4.2.5. Les différents états d'un système	84
4.2.5.1. Système fonctionnellement adéquat.....	84
4.2.5.2. Système coopératif	84
4.2.5.3. Système à milieu intérieur coopératif.....	84
4.2.6. Approche adoptée pour la démonstration de la propriété	85
4.3. Relations entre SC et SFA.....	86
4.3.1. $SC \subset SFA$	86
4.3.2. Surjection de $SC \rightarrow SFA$	88
4.4. Relations entre SMIC et SC.....	90
4.4.1. $SMIC \subset SC$	90
4.4.2. Surjection de $SMIC \rightarrow SC$	91

4.5. Relations entre SMIC et SFA.....	94
4.6. Conséquences	95
4.6.1. La coopération	95
4.6.2. Le mécanisme d'auto-organisation.....	96
4.6.3. Discussion.....	97

Chapitre 5 : L'algorithme auto-organisateur

5.1. Introduction.....	99
5.2. Les différentes situations non coopératives recensées	100
5.2.1. Notations	101
5.2.2. La situation non coopérative d'incompétence	102
5.2.3. La situation non coopérative d'ambiguïté.....	103
5.2.4. La situation non coopérative d'improductivité	103
5.2.5. La situation non coopérative d'inutilité	104
5.2.6. La situation non coopérative de concurrence	105
5.2.7. La situation non coopérative de conflit.....	105
5.3. La relaxation restreinte.....	107
5.3.1. Principe.....	107
5.3.2. Le paradigme de la relaxation restreinte.....	108
5.3.2.1. Association d'une durée de vie au message d'appel d'offre	109
5.3.2.2. Gestion du délai d'attente local.....	110
5.3.3. Propriétés.....	110
5.4. La communication spontanée	112
5.4.1. Principe.....	113
5.4.2. Mise en oeuvre	113
5.4.3. Propriétés.....	114
5.5. Discussion	116

PARTIE 3

117

Chapitre 6 : Présentation du projet ARCADIA

6.1. Introduction.....	117
6.2. Hypothèses du projet ARCADIA.....	118
6.2.1. Étude comparative entre la recherche d'information.....	119
6.2.2. Les systèmes d'information coopératifs.....	121

6.3. Approche adoptée.....	121
6.3.1. Arcadia : un système dans un environnement.....	122
6.3.2. Premiers éléments de réponse.....	122
6.4. Architecture fonctionnelle d'ARCADIA.....	125
6.4.1. Présentation des trois types d'agents.....	126
6.4.2. Les agents de médiations	126
6.4.3. Les agents de transactions	126
6.4.4. Les agents de croyances	127
6.4.5. Justification d'une telle architecture	127
6.4.6. ARCADIA : un système multi-agent auto-organisateur.....	128
6.5. Cadre des diverses démonstrations	128
6.5.1. Les différents mécanismes de communication utilisés.....	129
6.5.1.1. Structure des messages échangés entre AdM et/ou AdT.....	129
6.5.1.2. Structure des messages échangés entre AdCs.....	132
6.5.1.3. Mécanismes d'implantation des communications inter-agents .	133
6.5.2. Les différentes sources d'information utilisées.....	134
6.6. Discussion.....	135

Chapitre 7 : Le système multi-agent d'agents de médiation

7.1. Introduction.....	137
7.2. Structure d'un AdM	137
7.3. La communication chez l'AdM.....	138
7.3.1. Message en provenance d'un AdM.....	139
7.3.2. Message en provenance du timer.....	140
7.3.3. Message en provenance d'un AdT.....	140
7.4. L'interprétation chez l'AdM	141
7.4.1. Instanciation des situations non coopératives.....	141
7.4.2. Détection des situations non coopératives.....	142
7.4.2.1. Construction des deux tables de croyances.....	143
7.4.2.2. Mise en évidence de l'incompétence et de l'ambiguïté.....	143
7.4.2.3. L'incompétence partielle, la concurrence.....	144
7.4.3. Réactions face à ces situations non coopératives	145
7.4.3.1. Message sans destinataire en provenance d'un AdM	145
7.4.3.2. Message avec destinataire en provenance d'un AdM	146
7.4.3.3. Message en provenance d'un AdT	147
7.4.3.4. Message en provenance du timer	147
7.5. Le raisonnement chez l'AdM	147

7.6. Résultats obtenus à partir du système multi-agent d'AdMs.....	148
7.6.1. Mise en évidence de la communication inter-AdMs.....	148
7.6.2. Mise en évidence de la communication spontanée	149
7.6.3. Mise en évidence de la relaxation restreinte.....	150
7.7. Discussion	151

Chapitre 8 : Le système multi-agent d'agents de transaction

8.1. Introduction.....	153
8.2. Structure d'un AdT.....	153
8.3. La communication chez l'AdT	154
8.3.1. Message en provenance d'un AdM.....	155
8.3.2. Message en provenance d'un timer.....	155
8.3.3. Message en provenance d'un AdT	155
8.3.4. Message en provenance de son adaptateur.....	156
8.4. L'interprétation chez l'AdT.....	156
8.4.1. Instanciation des situations non coopératives	156
8.4.2. Détection des situations non coopératives	157
8.4.2.1. Construction des deux tables de croyances	157
8.4.2.2. L'incompétence, l'ambiguïté.....	158
8.4.2.3. La compétence partielle, la concurrence.....	158
8.4.3. Réactions face à ces situations non coopératives.....	159
8.4.3.1. Message sans destinataire en provenance d'un autre AdT ..	159
8.4.3.2. Message en provenance d'un AdT avec destinataire	159
8.4.3.3. Message en provenance d'un AdM.....	159
8.4.3.4. Message en provenance du timer.....	160
8.4.3.5. Message en provenance de l'adaptateur	160
8.5. Résultats obtenus a partir du système multi-agent d'AdTs	161
8.5.1. Mise en évidence de la relaxation restreinte.....	161
8.5.2. Mise en évidence de la communication spontanée	162
8.5.3. Mise en évidence de la combinaison de services	164
8.6. Discussion	165

Chapitre 9 : Le système multi-agent d'agents de croyances

9.1. Introduction.....	167
9.2. Typologie des agents.....	168

9.3. Fonctionnement du réseau d'AdC.....	169
9.3.1. Insertion d'une requête dans le SMA.....	169
9.3.2. Propagation du signal.....	170
9.3.3. Le feed-back des transactions AdT/AdM.....	170
9.4. Structure des agents de croyances.....	171
9.5. Communication chez les agents de croyances.....	171
9.5.1. La pondération des croyances	171
9.5.2. Calcul de la somme des poids associés aux agents activateurs	172
9.5.3. La dynamique d'un agent.....	172
9.5.4. Les trois états d'un agent.....	173
9.5.5. Contenu des messages échangés entre agents	174
9.5.6. La communication inter-AdCs	174
9.6. Mise en oeuvre de l'interprétation chez les AdCs	175
9.6.1. Instanciation des situations non coopératives	176
9.6.2. Détection et traitement des situations non coopératives.....	176
9.6.2.1. L'incompétence.....	177
9.6.2.2. L'improductivité.....	177
9.6.2.3. Le conflit.....	177
9.6.2.4. La concurrence.....	175
9.6.2.5. L'inutilité	178
9.7. Résultats obtenus à partir du système multi-agent d'AdCs.....	178
9.7.1. Proportion de messages non coopératifs.....	179
9.7.2. Étude d'agents particuliers	180
9.7.2.1. L'agent pavehicles	180
9.7.2.2. Les agents "from", "subject", "date".....	181
9.7.3. Nombre de termes par requête.....	182
9.7.3.1. Nombre de termes activés dans une requête	182
9.7.3.2. Termes supprimés dans les requêtes	183
9.7. Discussion.....	184

CONCLUSION ET PERSPECTIVES

185

1. Discussion et conclusion	185
2. Perspectives	188

BIBLIOGRAPHIE**189**

ANNEXES**201**

Annexe A : Exemples de systèmes reposant sur notre méthode d'apprentissage

A.1. Introduction.....	1
A.2. Application au jeu du tileworld.....	1
A.2.1. Détection et traitement des situations non coopératives.....	2
A.2.1.1. La situation non coopérative d'ambiguïté.....	2
A.2.1.2. La situation non coopérative de concurrence.....	3
A.2.1.3. La situation non coopérative de conflit.....	4
A.2.2. Résultats expérimentaux et analyse.....	5
A.3. Application aux fourragements par des fourmis.....	7
A.3.1. Détection et traitement des situations non coopératives.....	7
A.3.1.1. La situation non coopérative d'ambiguïté.....	7
A.3.1.2. La situation non coopérative de concurrence.....	8
A.3.1.3. La situation non coopérative d'improductivité.....	8
A.3.2. Résultats expérimentaux et analyse.....	9
A.4. Conclusion.....	10

Annexe B : Trace illustrant la communication spontanée inter-AdTs**Annexe C : Trace illustrant la relaxation restreinte inter-AdMs et inter-AdTs**

INTRODUCTION

1. CADRE DE L'ÉTUDE

"Depuis la création du premier outil, l'homme a toujours essayé de comparer le cerveau à la machine la plus perfectionnée du moment. Ainsi, pour les romains, la mémoire était un système d'égoûts très sophistiqué. Puis vint l'époque des grands standards téléphoniques, alors l'homme compara naturellement son cerveau à un standard téléphonique. Puis vint l'ordinateur et l'homme compara naturellement le cerveau à l'ordinateur. Mais c'est vers 1960 que l'homme a abordé le problème suivant : créer des comportements "intelligents" à partir d'ordinateurs. C'était le début de l'Intelligence Artificielle" [Perez, 1988].

Depuis, l'intelligence artificielle s'intéresse à la modélisation des comportements humains suffisamment complexes pour être dits intelligents. Turing proposa un test consistant à dire qu'une machine est intelligente si on ne peut la distinguer d'un humain lors d'une conversation. Cependant, la notion d'intelligence de groupe ou d'intelligence collective était totalement inexistante dans ce test, car il n'était pas demandé à la machine de se mettre à la place de plusieurs personnes.

L'objet de notre étude est justement d'analyser et de recréer l'aspect dynamique du comportement inhérent à un groupe d'entités. Dans une société humaine, par exemple, un problème complexe est très souvent résolu par un groupe d'individus compétents dont l'organisation et l'attitude évoluent dynamiquement et de manière autonome au cours de l'élaboration de la solution.

C'est cette dynamique de l'organisation que nous proposons de reproduire à travers les concepts de l'intelligence artificielle distribuée et des systèmes multi-agents.

L'Intelligence Artificielle Distribuée (couramment notée IAD) se différencie de l'Intelligence Artificielle (IA) classique non seulement par le concept de distribution de l'intelligence, mais aussi par un paradigme différent. A la différence de l'intelligence artificielle, la connaissance n'est plus concentrée dans une seule entité, mais répartie entre plusieurs, qui, par leurs interactions, arrivent à résoudre une tâche collective ou à coordonner leurs actions.

Les **Systèmes Multi-Agents (SMA)**, quant à eux, considèrent comme fondement de leur théorie que “les activités simples ou complexes, telles que la résolution de problèmes, l’établissement d’un diagnostic, la coordination d’action ou la construction de systèmes sont le fruit d’entités relativement autonomes et indépendantes, appelées agents, qui travaillent au sein de communautés selon des modes, parfois complexes, de coopération, de conflit et de concurrence, pour survivre et se perpétuer. De ces interactions émergent des structures organisées qui, en retour, contraignent et modifient les comportements de ces agents.” [Ferber, 1995]

Les systèmes multi-agents sont composés d’un agrégat de logiciels autonomes appelés “agents” et sont caractérisés par :

- un environnement dynamique dans lequel sont plongés les agents. L’activité de ces derniers peut ainsi être remise en cause durant la vie du système ;
- des agents ayant une perception partielle de leur environnement. La pertinence de leur action est alors uniquement locale ;
- des agents ayant des capacités cognitives restreintes. Ils ne peuvent, par conséquent, pas tout traiter individuellement et doivent s’associer pour réaliser l’activité globale.

Comme le souligne Jacques Ferber [Ferber, 1995], les recherches dans le domaine poursuivent deux objectifs majeurs:

- le premier concerne l’analyse théorique et expérimentale des mécanismes d’auto-organisation qui ont lieu lorsque plusieurs entités autonomes interagissent ;
- le second s’intéresse à la réalisation d’artefacts distribués capables d’accomplir des tâches complexes par coopération et interaction.

Ce travail de thèse approfondit plus particulièrement le premier point, à savoir l’auto-organisation dans les systèmes multi-agents fondée sur l’analyse coopérative de leurs interactions.

2. LA PROBLÉMATIQUE ÉTUDIÉE

Les investigations actuelles dans le domaine multi-agent s’intéressent particulièrement aux systèmes constitués de nombreux agents et ayant une très forte dynamique. Chaque agent n’a qu’une vue partielle du système dans lequel il est situé et évolue de manière autonome. Il est donc impossible d’avoir, dans de tels contextes, une vue globale du système. D’autre part, la dynamique étant très importante il est impossible de prévoir, dès la conception, toutes les situations pouvant survenir durant le fonctionnement du système. Ces systèmes doivent donc s’adapter à cette dynamique au cours de leur fonctionnement.

Un système multi-agent n'est pas monolithique mais constitué au contraire d'un grand nombre d'agents interagissant les uns avec les autres. Chaque agent agit sur l'environnement, sur autrui et sur l'organisation en fonction de ses interactions.

La résolution d'un problème s'effectue le plus souvent de manière descendante. Le problème est décomposé en sous-problèmes à résoudre individuellement et généralement en séquence pour pouvoir atteindre la solution cherchée. Cette approche présuppose de pouvoir maîtriser, dès la conception, les interactions potentielles du système avec l'environnement dans lequel il est plongé. Cette démarche devient difficilement applicable lorsque la spécification du problème est incomplète ou lorsque l'environnement est évolutif. Les systèmes multi-agents répondent partiellement à ce problème.

En effet, les systèmes multi-agents permettent de construire des systèmes artificiels pour lesquels l'activité collective observée n'est décrite dans aucun des agents qui les composent. Cette approche est nécessaire lorsqu'il s'agit de construire des systèmes complexes où il n'existe pas de solution algorithmique connue *a priori* : la solution doit se construire dynamiquement. Ceci est très utile dans les applications où :

- la coopération d'une multitude de tâches élémentaires est indispensable pour la réalisation d'une activité collective précise mais difficilement formalisable,
- le système est fortement évolutif : la création, la modification ou la suppression de tâches élémentaires sont permanentes et imprévisibles,
- l'environnement exerce des contraintes fortes et dynamiques sur le système.

Cette thèse s'intéresse tout particulièrement aux problèmes d'adaptation dynamique des agents. Notre objectif est de faire en sorte que dans un tel contexte, le système puisse aboutir à un comportement collectif cohérent alors que les agents cherchent à atteindre leurs objectifs individuels et qu'il n'existe nulle part dans le système un contrôle de leur activité fondé sur l'évaluation de la fonction globale à réaliser.

Nous cherchons à spécifier les actions pertinentes que des agents peuvent réaliser face à des situations imprévues tout en faisant abstraction de la sémantique particulière de ces situations et actions pour conserver toute leur généralité.

Nous nous attacherons à montrer les propriétés globales d'un système contenant des agents ayant de telles interactions locales.

3. LES PRINCIPALES DÉFINITIONS EMPLOYÉES

Ce paragraphe présente notre point de vue sur un certain nombre de notions de base employées dans ce travail. Elles peuvent être précisées par certains auteurs dans l'état de l'art, mais seront surtout employées à partir du chapitre 4.

L'environnement d'un système

Nous allons expliciter à la fois la notion d'environnement pour un agent et celle pour un système. L'environnement d'un agent est tout ce qui est extérieur à l'agent. On distingue l'environnement dit "social" c'est-à-dire les agents qu'il connaît, et l'environnement dit "physique" constitué des ressources matérielles présentes dans le champ de perception de l'agent ou de ses propres effecteurs. Selon les caractéristiques des agents constituant le système, l'environnement physique a plus ou moins d'importance. Dans le cadre d'une société d'agents dits "logiciels" il est souvent négligé ; les problèmes qui en découlent sont supposés être résolus à un autre niveau et ne nous incombent alors plus (un message est supposé toujours arriver physiquement à son destinataire). En revanche, si l'on étudie un système multi-agent composé d'agents "matériels" (robots, agents se déplaçant sur un damier en quête de nourriture), l'environnement physique a une très grande importance.

Tout système possède également un environnement duquel il ne peut être dissocié. Cet environnement est constitué de tout ce qui est extérieur au système (notamment d'autres systèmes). Pour illustrer nos propos, supposons un système S qui est en fait l'IRIT et l'université Paul Sabatier qui est le monde W dans lequel S est situé. L'environnement de S est donc l'université Paul Sabatier privé de l'IRIT (ce que nous pouvons exprimer en notation ensembliste $C_{W,S}$), comprenant lui-même le système S' qui est par exemple l'administration, S'' qui est le gymnase etc...

Un système perçoit partiellement l'environnement dans lequel il est plongé (dans notre exemple, S peut connaître S' - et cela est même souhaitable - mais par contre il peut ne pas connaître S'') et interagit avec lui. Comme nous le verrons au paragraphe 3.3.1., la fréquence de ces échanges n'est pas la même selon que le système est qualifié d'ouvert ou de fermé.

L'organisation d'un système

L'organisation d'un système, désigne les liens possibles (interaction, accointance...) entre les agents qui composent le système [Chevrier, 1996]. Elle est donc représentée par un réseau de relations d'accointances, dans lequel, les agents correspondent aux noeuds et les arcs aux différentes interactions possibles entre les agents reliés. Au tout début du fonctionnement d'un système, les agents construisent leurs accointances en fonction de la topologie du réseau existant. Si le réseau est vide, le système est totalement inorganisé, auquel cas il est construit en fonction des événements perçus et traités. Cette définition sera approfondie au paragraphe 3.3.2.

L'auto-organisation d'un système

Étant donnée notre définition de l'organisation, l'auto-organisation d'un système correspond à la réorganisation dynamique, et décidée de manière autonome, des liens entre les agents qui le constituent. Ce changement d'organisation en fonction d'une contrainte imposée par l'environnement amène le système à s'adapter et à fournir une réponse qui peut diverger de celle précédemment transmise.

Système fonctionnellement adéquat

Un système est fonctionnellement adéquat s'il réalise la "bonne" fonction dans son environnement. La signification de "bonne" fonction est utilisée ici au sens de Maturana lorsqu'il dit "tout ce qui est dit est dit par un observateur" [Maturana, 1980]. L'activité correcte n'est donc décidable que par un observateur extérieur jugeant les interactions et connaissant la fonction que le système doit réaliser dans son environnement.

Système coopératif

Nous dirons qu'un système est coopératif s'il est en permanence en interactions coopératives avec son environnement. Nous préciserons cette définition très générale au chapitre 4.

Milieu intérieur d'un système

Un système est composé de plusieurs composants (structures physiques ou processus qui sont utilisés pour implémenter un mécanisme). Le milieu interne du système est l'ensemble de ses composants et des supports physiques utilisés pour les échanges internes. Dans un système multi-agent, il s'agit des agents présents dans le système, des accointances qui les relient ainsi que des moyens de communication.

4. PRÉSENTATION DE LA THÈSE

Cette thèse propose une méthode d'adaptation dynamique indépendante du domaine d'application. Nous avons, pour cela, élaboré des techniques de coopération tendant à éviter les dysfonctionnements comme les conflits, la concurrence... L'imprévu étant inhérent à ce type de situation, nous faisons le postulat suivant : pour faire "converger" un système artificiel alors que ses attracteurs ne sont pas connus *a priori*, il est nécessaire :

- de donner aux agents artificiels le moyen d'agir dans leur environnement et avec autrui,
- de considérer le système comme un ensemble de parties (agents) en synergie, toutes indispensables à l'activité collective,
- de spécifier les comportements d'un "agent coopératif" générique, indépendamment du domaine d'application, pour réaliser des systèmes multi-agents véritablement **auto-organiseurs**.

Dès lors, il ne devient plus nécessaire de connaître la finalité d'un système particulier avant sa conception, mais de savoir que toutes ses parties doivent **coopérer**. Faire coopérer effectivement des agents dans les systèmes artificiels, comme nous l'entendons, revient donc à avoir un collectif qui réalise la "bonne" fonction globale. Le contrôle de la résolution au sens habituel du terme est alors supprimé [Glize, 1996]. Le contrôle usuel correspond à des "activités du système et de répartition des rôles et tâches parmi les agents afin que le système aboutisse à une solution" [Chevrier, 1996]. Il s'agit ici de faire en sorte que la coopération soit omniprésente entre les parties constituant le système car c'est elle qui réalise le contrôle.

Les agents ne connaissent donc rien de la finalité du système. Leur unique but est de rester en situation coopérative avec les agents qu'ils connaissent tout en s'adaptant à la dynamique inhérente au système (en s'auto-organisant). Chaque agent hérite alors (par "encapsulation") de connaissances génériques de coopération. Le comportement collectif n'est donc pas explicitement programmé dans chaque agent. Il émerge par effet de bord des comportements sociaux et des

croyances que possèdent chaque agent ainsi que des interactions qu'ils réalisent les uns avec les autres. "L'organisation globale émerge donc du système des interactions locales" [Lenay, 1994].

Le manuscrit de cette thèse se divise en trois parties.

- Une première partie, appelée état de l'art, présente les trois domaines de recherche que nous abordons, à savoir l'apport de la coopération et l'apprentissage dans le domaine multi-agent ainsi que l'auto-organisation (dans les domaines de la vie artificielle, de la systémique, des réseaux neuronaux, de la vie artificielle, de la robotique et de l'IAD). Notre méthode d'auto-organisation ne fait aucun présupposé ni contrôle sur l'organisation attendue : cette dernière doit se construire et s'auto-modifier dynamiquement. C'est la raison pour laquelle nous n'avons volontairement pas cité dans cet état de l'art de travaux concernant la théorie sur les organisations. De même, notre méthode ne repose sur aucune notion sémantique ; les messages échangés entre agents ne sont considérés que d'un point de vue syntaxique. Nous n'avons par conséquent mentionné dans cet état de l'art aucun travail ayant trait aux intentions, engagements (Cohen et Levesque, Georgeff, ...)

- La deuxième partie présente la méthode d'apprentissage par auto-organisation proposée ainsi qu'une formalisation de celle-ci. Elle repose sur la théorie suivante : si les entités composant un système entretiennent des relations coopératives les unes avec les autres et s'adaptent à la dynamique inhérente au système (dynamique qui fait parfois basculer le système dans une situation qualifiée de non coopérative car dégradant l'activité collective de la société) tout en apprenant, le système fournit une solution satisfaisante au problème en cours de résolution. L'apprentissage porte essentiellement sur l'organisation : l'agent doit être situé au bon endroit et au bon moment dans l'organisation. Les compétences sont également modifiées par apprentissage sur l'organisation, sachant que nous aurons des systèmes multi-agents "imbriqués" évoluant selon les mêmes critères. Ces systèmes multi-agents seront des systèmes auto-organiseurs visant à s'adapter à toutes les situations perturbatrices en tentant de revenir à une situation coopérative.

- La dernière partie présente l'instanciation de notre méthode au domaine de la recherche d'information dans un contexte distribué et fortement dynamique. Nous avons validé la méthode que nous préconisons sur une architecture appelée "ARCADIA" (ARchitecture pour la Coopération d'Agents D'Information Autonomes) dans le cadre d'un contrat CNET. Le but de ce travail est d'élaborer une méthode basée sur des techniques multi-agents, permettant d'**accéder à de l'information pertinente** dans un réseau de ressources d'information distribuées géographiquement, hétérogènes de par leur contenu et en constante évolution. L'objectif de ce projet est donc de permettre aux utilisateurs d'ARCADIA, d'accéder à des services commerciaux électroniques de manière simple et efficace via des opérateurs de communication comme France Télécom. Les principes de base de l'auto-organisation ont été appliqués dans le cadre de cette étude. Chaque service ou usager abonné à ARCADIA est respectivement représenté par un agent qui agit de manière autonome pour le compte de l'utilisateur ou du service qu'il représente c'est-à-dire

sans que ce dernier ait besoin de le surveiller. Son but propre est de résoudre la requête qui lui a été assignée tout en restant coopératif avec les agents du système avec qui il interagit. Un utilisateur abonné à ARCADIA et cherchant un renseignement particulier, émet une requête en langage naturel. Le système doit alors trouver par lui-même le (ou les) service(s) le(s) plus approprié(s) à la demande si toutefois il(s) existe(nt). Il pourra être éventuellement amené à faire des combinaisons de services.

Cette partie présente également la spécification des trois types d'agents que nous avons définis dans le cadre du projet ARCADIA ainsi que les résultats obtenus suite aux différents prototypes réalisés.

CHAPITRE 1

LA COOPÉRATION DANS LES SYSTÈMES MULTI-AGENTS

1.1. INTRODUCTION

La coopération nous paraît être une caractéristique essentielle dans l'apprentissage en univers multi-agent. En effet, chaque agent ne possède qu'une vue partielle du système auquel il appartient et il a la possibilité, grâce à la coopération, de réaliser plus de tâches que s'il avait travaillé seul de manière individualiste.

Les travaux les plus anciens concernant la coopération utilisent la coopération comme processus de résolution de problèmes. Les chercheurs se sont ensuite plus particulièrement intéressés à la définition et à la formalisation de ce concept [Woolridge, 1995]. Le concept de coopération prend une signification plus large dès lors qu'on l'utilise pour avancer qu'un système se trouve dans une situation coopérative. Il signifie toujours coopération pour la résolution du problème auquel est confronté le système, mais aussi coopération entre les entités du système, en l'occurrence les agents, pour que le système ait un fonctionnement optimal. Nous nous plaçons dans ce second cas pour étudier la coopération.

Nous souhaitons montrer, que l'activité coopérative mais aussi l'attitude sociale coopérative permettent d'optimiser le fonctionnement du collectif par l'adaptation. Plusieurs exemples peuvent illustrer la notion de coopération pour un agent, au sens où nous l'entendons :

- envoyer spontanément une information aux autres agents, s'il croit qu'elle peut leur être utile,
- accepter de satisfaire une demande explicite si elle n'est pas en contradiction avec son activité courante ou ses objectifs,
- agir sur le monde s'il croit qu'il peut favoriser les autres, même s'il n'en retire pas de satisfaction personnelle.

Ce chapitre se compose de quatre paragraphes. Le premier a trait à la terminologie : il présente différentes définitions de la coopération, de la coordination et de la collaboration. Le second paragraphe montre l'apport de la coopération dans divers exemples. Le troisième étudie la

coopération comme attitude sociale. Enfin, le dernier paragraphe est une discussion sur les divers articles présentés.

1.2. CADRE DE L'ÉTUDE

L'intérêt des systèmes multi-agents ne porte pas sur la manière dont un agent résout un problème donné mais sur la manière dont un groupe d'agents arrive à résoudre ce problème. En effet, une société ne doit pas seulement être vue comme une agrégation d'agents interconnectés (directement ou indirectement par le biais de l'environnement) ayant des compétences particulières (homogènes ou hétérogènes). Une société repose au contraire sur une réelle coopération des agents les uns avec les autres et non sur une simple cohabitation. L'aspect collectif lors de la résolution d'un problème et plus exactement la coopération des agents sont donc des points importants dans ce domaine.

La coopération permet en particulier à des agents de résoudre des tâches qu'ils n'auraient pas pu résoudre seuls, d'améliorer la productivité de chacun, d'optimiser l'utilisation des ressources, d'augmenter le nombre de tâches réalisées lors d'un délai imparti... [Ferber, 1994].

Les exemples que nous allons présenter vont dans ce sens : nous montrerons comment la coopération a été mise en oeuvre dans les systèmes proposés et quelles sont les caractéristiques qui ont été accrues grâce à la coopération.

1.2.1. La coopération

Nous allons donner quelques définitions de la coopération, la coordination et la collaboration, ces notions étant parfois difficiles à différencier [Ferber, 1995], [Brassac, 1996], [Lenay, 1996].

La définition proposée par Petit Robert est la suivante : "action de participer à une oeuvre commune (collaboration, aide, contribution)".

Yves Demazeau et Jean-Pierre Müller [Demazeau, 1990] parlent de coopération pour une tâche locale, lorsqu'un agent a besoin de coopérer avec autrui parce qu'il n'est pas capable de l'accomplir par lui-même ou parce que les autres peuvent l'accomplir de manière plus efficace que lui (dans un intervalle de temps plus court).

Pour Jacques Ferber [Ferber, 1995], selon le niveau d'observation où l'on se place, la coopération revêt deux dimensions :

- la coopération peut être une attitude des agents qui décident de travailler en commun. Dans ce cas, les agents coopèrent s'ils s'engagent dans une action commune après avoir identifié et adopté un but commun (prise de conscience de l'existence d'un but commun et engagement de chacun),
- dans le second cas, un observateur interprète *a posteriori* les comportements en les qualifiant de coopératifs ou non, à partir de critères sociaux comme l'interdépendance des actions ou le nombre de communications effectuées.

Il considère que plusieurs agents coopèrent, pour atteindre un objectif commun, “si l'une des deux conditions est vérifiée :

- 1 - l'ajout d'un nouvel agent accroît différentiellement les performances du groupe,
- 2 - il existe des conflits potentiels ou actuels d'accès à des ressources et l'action des agents sert à éviter ou à sortir de tels conflits.”

Christian Brassac et Sylvie Pesty [Brassac, 1996] insistent sur la volonté individuelle et collective de coopérer chez les agents cognitifs. En revanche, la coopération n'apparaît, selon eux, au niveau des agents réactifs que parce qu'elle est attribuée par un observateur. Le terme coopération n'est selon eux pas adapté aux agents réactifs ; ils lui préfèrent la notion de co-action.

Charles Lenay [Lenay, 1996], définit la coopération comme “un type de dynamique collective qui aboutit à un état émergent bénéfique pour la population”. Il distingue les termes coopération et co-action de Ch. Brassac et S. Pesty par coopération intentionnelle et coopération non intentionnelle.

Ces différentes définitions s'entendent sur le fait qu'il existe deux points de vue possibles : le premier cherche à qualifier l'attitude des agents qui décident de travailler en commun et le second tente de qualifier l'activité d'un ensemble d'agents vue par un observateur extérieur qui n'aurait pas accès aux états mentaux des agents. Ce second point de vue est plus général et englobe le premier. Il nous semble donc nécessaire de nommer ce phénomène général plutôt que de réserver le terme de coopération uniquement aux agents cognitifs. Qu'ils soient cognitifs ou réactifs les agents artificiels n'ont pas l'intention de coopérer ; ils sont le reflet du comportement que leur a attribué le concepteur, ce comportement pouvant être qualifié, d'un point de vue extérieur, de coopératif.

1.2.2. La coordination

La coordination est définie par le Petit Robert de la manière suivante : “agencement des parties d'un tout selon un plan logique, pour une fin déterminée (organisation, coordination des opérations d'une troupe).

“Pour garantir la cohérence globale d'un SMA et sa convergence vers une solution au problème posé, il est nécessaire que les agents coordonnent leurs différentes actions. Cette coordination sous-entend que les agents doivent négocier, décider de leur rôle respectif, échanger des informations pertinentes mais aussi construire et gérer une représentation de leur environnement et des autres agents [Haton, 1996]”.

Selon Thomas Malone, la coordination d'actions est "l'ensemble des activités supplémentaires qu'il est nécessaire d'accomplir dans un environnement multi-agent et qu'un seul agent poursuivant les mêmes buts n'accomplirait pas" [Malone, 1988].

Selon Durfee et Montgomery [Moraitis, 1994], “le but de la coordination est de trouver, parmi un ensemble de comportements d’agents qui interagissent, une collection de comportements qui réalise d’une façon satisfaisante les objectifs les plus importants des agents. Le degré de coordination reflète dans quelle mesure les agents peuvent éviter des activités accessoires afin d’arriver à leurs fins essentielles.”

Michelle Courant [Brassac, 1996] énonce “il y a coordination dès que deux entités sont impliquées dans la création/émergence d'un ordre commun au sens physique.”

Pour Thomas Malone et Michelle Courant, la coordination subsume les mécanismes d’interdépendances alors que pour E. Durfee et Jean Paul Haton, la coordination d'actions est considérée comme un moyen de résoudre un ensemble de sous problèmes sous-entendus par la répartition de tâches.

Dans le cadre de sociétés composées d'agents autonomes qui poursuivent des objectifs individuels, la coordination est une méthode indispensable pour agencer de manière cohérente les interventions de chacun. Elle définit une relation d'ordonnancement et de dépendance entre les actions. Elle est nécessaire :

- lorsque les agents ont besoin d'informations et de résultats que seuls d'autres agents peuvent fournir,
- lorsque les ressources sont limitées. Il est alors nécessaire de coordonner les attitudes des agents pour éviter les éventuelles collisions d'accès,
- pour optimiser les coûts en éliminant les actions inutiles et en évitant les redondances d'actions,
- et enfin pour permettre à des agents ayant des objectifs distincts mais dépendants les uns des autres, de satisfaire ces objectifs et d'accomplir leur travail en tirant éventuellement parti de cette dépendance.

Dans les systèmes multi-agents réactifs, la coordination d'actions fait essentiellement appel à des techniques de champs de force et de marquage de l'environnement. Nous penchons donc plutôt pour la définition de la coopération fournie par E. Durfee et Jean Paul Haton.

1.2.3. La collaboration

Le Petit Robert propose la définition suivante pour la collaboration : “travail en commun, action de collaborer avec quelqu'un (coopération, aide, association).”

Selon Jacques Ferber [Ferber, 1994], la collaboration indique “l'ensemble des techniques permettant à des agents de (se) répartir des tâches, des informations et des ressources de manière à réaliser une oeuvre commune”.

Yves Demazeau et Jean-Pierre Müller [Demazeau, 1990] parlent de collaboration pour des tâches globales, et plusieurs agents prenant part à la collaboration sont capables d'effectuer individuellement chacune de ces tâches.

La collaboration fait unanimement référence à la résolution d'une tâche globale par plusieurs agents. Cette aptitude ne fait pas partie de nos préoccupations puisqu'elle présuppose de connaître la fonction globale pour laquelle les agents doivent collaborer.

1.3. ÉTUDE DES DIFFÉRENTS TRAVAUX

Un système multi-agent peut être considéré (d'un point de vue statique) comme un ensemble de compétences distribuées. Il est donc légitime de croire au premier abord que l'efficacité du travail collectif d'une société d'agents est proportionnelle à la quantité d'agents ayant pris part à la résolution. Ce n'est pourtant pas ce qui a été observé en pratique par Mataric [Mataric, 1994] en robotique, par Genesereth [Genesereth, 1994] et Etzioni [Etzioni, 1994] en recherche d'information. Ce phénomène provient du fait que des agents trop individualistes provoquent d'autant plus de situations indésirables pour la collectivité (conflits, concurrences,...) qu'ils sont nombreux. Dès lors, on comprend bien l'intérêt de la coopération.

Ce paragraphe présente des travaux essentiels qui sont classés en deux grandes familles : celle qui présente l'apport de la coopération dans la résolution de problème et celle qui considère la coopération comme attitude sociale d'un agent.

1.3.1. L'apport de la coopération pour la résolution

Nous allons présenter dans ce paragraphe les travaux de quatre auteurs qui montrent l'apport de la coopération dans la résolution de problème.

1.3.1.1. Les travaux de Bouron & Ferber

Les trois grandes fonctions de la coopération définies par Jacques Ferber sont l'accroissement des performances, la résolution de conflits et l'amélioration de la survie. La première fonction se mesure de manière qualitative et quantitative tant d'un point de vue individuel que collectif. Les deux dernières fonctions reflètent la capacité d'adaptation d'un individu ou d'un groupe lui permettant de maintenir son intégrité fonctionnelle. Ceci est particulièrement utile en monde ouvert. Une société d'agents autonomes sera amenée à faire face aux éventuelles perturbations ou aux éventuels dysfonctionnements tendant à la détériorer.

Les mécanismes sociaux mis en place pour assurer la viabilité du groupe restreignent encore les possibilités individuelles des agents. Ils contraignent leurs comportements et les poussent à se socialiser encore plus c'est-à-dire à devenir de plus en plus spécialisés et dépendant des autres. "Les organisations émergent des interactions sociales entre individus et contraignent en retour leur comportement" [Ferber, 1994].

Thierry Bouron [Bouron, 1992] a défini des indices de coopération permettant de mesurer les activités de coopération entre agents. Il s'agit du degré de parallélisation, de la coordination d'actions, de la robustesse, de la non redondance des actions, de la non persistance des conflits et enfin du partage des ressources. Ces indices permettent de s'affranchir des caractéristiques internes des agents et de ne s'occuper que de leur comportement observable. Leurs définitions reposent d'ailleurs sur des considérations d'observabilité et de quantifiabilité.

1.3.1.2. La méthode Goldman & Rosenschein

Claudia Goldman et Jeffrey Rosenschein [Goldman, 1994] ne s'interrogent pas sur la manière dont un agent doit contrôler son propre comportement pour satisfaire un but et être coopératif avec autrui, mais sur la manière dont un agent pourrait transformer le monde de manière coopérative, en même temps qu'il poursuit son propre but. Un comportement coopératif consiste alors pour un agent à modifier l'environnement de manière individuelle et indépendante pour qu'il soit propice au

fonctionnement des autres agents. Un comportement coopératif peut consister par exemple à poser un objet plus loin sur le sol, sachant qu'un autre agent sera ainsi capable de le trouver plus facilement.

Leur travail a été double : il a consisté d'une part à vérifier qu'un comportement coopératif chez chaque agent améliorerait le comportement collectif, puis à établir des règles génériques de coopération en fonction du domaine étudié. Ils se sont ensuite intéressés à la généralisation de leur méthode : la construction de règles de coopération génériques.

Leurs réflexions ont été instanciées avec le jeu du tileworld, jeu dans lequel des agents se déplacent sur un damier 10x10 en évitant les obstacles éventuels (barrières) et en poussant la tuile la plus proche d'eux (une à la fois) dans le trou le plus proche.

Ils ont, dans un premier temps, étudié quel effet la coopération, au sens où ils l'entendent, pouvait avoir sur le comportement collectif des agents du tileworld. Les résultats s'étant avérés très positifs, ils ont cherché à définir une méta-règle permettant aux agents, dans le cadre du tileworld, de s'aider mutuellement, sans avoir à chercher un plan optimal. Cette méta-règle est la suivante : les tuiles doivent avoir un degré de liberté très important, c'est-à-dire être librement déplaçables et moins contraintes par des barrières. Chaque tuile possède un degré de liberté qui correspond au nombre de directions dans lesquelles elle peut être poussée. Ainsi, lorsqu'un agent désirant pousser une tuile dans un trou donné trouve une tuile ayant peu de degré de liberté proche de son passage, il va faire un détour pour essayer d'augmenter le degré de liberté de celle-ci. Ce type de coopération ne nécessite pas que les agents communiquent, ni qu'ils aient un modèle des plans ou des buts spécifiques des autres agents (puisque chaque agent est supposé connaître l'état courant de la grille). La coopération va émerger du comportement social des agents individuels.

Étant donné le gain d'efficacité comparé au peu de difficulté pour sa mise en oeuvre, ils ont immédiatement cherché à généraliser leur méthode indépendamment du domaine d'application. Ils décident pour cela de caractériser le domaine par un automate fini déterministe et les règles de coopération par des transformations de cet automate. Trois règles coopératives de changement d'état sont proposées et correspondent à des transformations de l'automate. Dès lors, les concepteurs de système doivent identifier les états globaux qui sont désirables (dans le tileworld la liberté des tuiles) et appliquent ensuite les règles de changements définies qui permettent aux agents d'effectuer des transformations d'états qui tendent vers l'état social désiré. La coordination émerge alors des agents qui utilisent cette règle de coopération sans coordination ou négociation explicite.

1.3.1.3. La méthode de Prasad, Lander, Lesser

Nagendra Prasad [Prasad, 1996] propose deux algorithmes d'apprentissage qui visent à supprimer les problèmes engendrés par le fait que les agents ne possèdent qu'une vue partielle de

l'espace de recherche et par la même éviter les conflits inhérents à cette lacune. Ces algorithmes ont été insérés dans le projet STEAM qui est un système de recherche distribuée dans lequel chaque agent effectue une recherche locale basée sur des contraintes explicites et implicites contenues dans son contexte local. Dans STEAM, une recherche est effectuée sur un espace de modèles partiels. Un modèle partiel représente une solution partielle dans l'espace de recherche global (composite). Elle est initiée en plaçant une spécification du problème dans une mémoire partagée centralisée, qui agit comme un répertoire pour les solutions partielles et qui est visible par tous les agents. Certains des agents initient des propositions en fonction des spécifications du problème, de leurs contraintes locales et de leur état local. Les autres agents, à leur tour, étendent ou au contraire critiquent ces propositions pour former des modèles complets. Un agent peut donc jouer trois rôles : initiateur, développeur ou critique d'un modèle. Ces trois actions peuvent être qualifiées de coopératives dans le sens où les agents s'apportent une aide mutuelle pour avoir une vue plus exacte de l'espace de recherche.

Le premier algorithme d'apprentissage proposé s'appelle CDL ("Conflict Driven Learning"). Lorsqu'un agent désire étendre ou critiquer un modèle partiel, il doit détecter que le modèle transgresse certaines de ses contraintes locales. Ses contraintes peuvent être soit explicites soit implicites. Les contraintes explicites peuvent être partagées, auquel cas l'agent qui détecte des transgressions de contraintes explicites génère un feed-back aux agents à l'origine du modèle partiel qui a généré le conflit. Un tel feed-back permet aux agents de développer une approximation de l'espace global de recherche qui inclut à la fois leur perspective locale (qui leur est propre) et une contrainte explicite qui est construite à partir des autres agents, ce grâce à la coopération.

Le second algorithme d'apprentissage est appelé CBL ("Case Based Learning"). Durant la phase d'apprentissage, les agents effectuent leur recherche en utilisant CDL. A la fin de chaque recherche, un agent mémorise la spécification du problème et les contraintes non locales qu'il a reçu par feed-back des autres agents. Ainsi, lorsqu'une nouvelle instance de problème est présentée à l'ensemble des agents, ce dernier retrouve l'ensemble des contraintes non locales qui sont sauvegardées et liées à la spécification d'un problème antérieur qui est similaire à la spécification du problème actuel. Il les ajoute à l'ensemble des demandes locales au début de la recherche. Ainsi les agents peuvent éviter des communications pour atteindre des approximations de l'espace de recherche global.

Trois stratégies de recherche ont été expérimentées :

- la recherche à l'aveugle dans laquelle aucun apprentissage n'est appliqué. Lorsqu'un agent détecte un conflit dans un modèle particulier, il choisit un autre modèle pour poursuivre son raisonnement. Les agents ne communiquent pas d'information,
- l'apprentissage orienté conflit : un agent qui détecte un conflit génère un feed-back qui est assimilé par les agents récepteurs,

- l'apprentissage orienté cas : les agents utilisent les cas accumulés précédemment pour commencer leur résolution de problème avec une connaissance des contraintes non locales.

Les résultats obtenus montrent que les deux algorithmes améliorent à la fois la qualité de la solution et le temps de calcul par rapport à une recherche à l'aveugle. De plus, une fois que l'apprentissage est terminé, la méthode CBL diminue le temps de communication.

1.3.1.4. Les travaux de Hogg & Huberman et Ebstyne

[Hogg, 1993] et [Ebstyne, 1996] ont montré que la coopération permettait d'accélérer le processus de résolution d'un problème. Cette coopération implique une collection d'agents qui interagissent en communiquant des informations ou des suggestions tout en résolvant un problème. Ils illustrent leur propos de la manière suivante : dans un algorithme génétique, les membres d'une population d'états échangent des parties d'eux-mêmes ou mutent pour créer une nouvelle population contenant souvent des états ayant une valeur importante. Un autre exemple est un réseau neuronal, où la sortie d'un neurone affecte le comportement du neurone la recevant.

Dans leurs expériences, toutes les suggestions sont écrites dans un tableau noir centralisé. Chaque agent peut ainsi accéder aux résultats d'un autre agent. Le but de leur travail a été d'étudier l'apport de la coopération au domaine de la cryptarithmétique.

La cryptarithmétique cherche une unique attribution de chiffre à chaque lettre du monde pour que les nombres représentés par les mots soient additionnés correctement (DONALD + GERALD = ROBERT si A=4, B=3, D=5, E=9, G=1, L=8, N=6, O=2, R=7, T=0). En général, s'il y a n lettres il y a 10^n états possibles. Plutôt que de chercher des heuristiques intelligentes, ce qui nécessite beaucoup d'énergie, ils préfèrent utiliser des règles de coopération simples permettant d'atteindre rapidement la solution. Ils utilisent une "suggestion" qui consiste à des attributions lettre/chiffre pour qu'elles s'additionnent correctement par colonne ($AB + AC = DE$ est valable pour la seconde colonne car on a bien $B+C=E$). Ces suggestions sont postées dans un tableau noir. Aussi, une recherche coopérative permet à l'agent d'utiliser des suggestions trouvées par d'autres grâce au tableau noir. A chaque étape, un agent choisit aléatoirement une suggestion dans le tableau noir (s'il y en a) et remplace les attributions dans son état courant par celles spécifiées dans la suggestion. Une fois que l'agent obtient le nouvel état, il génère et poste à son tour toutes les suggestions possibles à partir de son état, s'il y en a. Ainsi, les attributions qui marchent pour plus d'une colonne sont postées comme différentes suggestions.

Cette application a permis de montrer que plus le problème croît en difficulté, plus l'importance de la coopération et l'utilité d'une mémoire à accès rapide sont augmentées. Après avoir testé leur méthode de coopération, ils sont arrivés à la conclusion suivante : plutôt que de dépenser de

l'énergie à développer un programme monolithique ou une heuristique parfaite, il est meilleur d'avoir un ensemble de processus coopérant de manière relativement simple qui travaillent de manière concurrente sur le problème tout en communiquant leurs résultats partiels.

1.3.2. La coopération en tant qu'attitude sociale

Les trois études que nous allons présenter considèrent la coopération comme attitude sociale chez les agents. Elles définissent plusieurs attitudes sociales, parmi lesquelles la coopération, et étudient les différences de comportements et les avantages de chacune d'entre elles.

1.3.2.1. Les travaux de Cesta, Miceli, Rizzo

Les travaux de Amedeo Cesta et al. [Cesta, 1996] étudient l'effet de différentes attitudes d'interactions sur la survie d'agents en quête de nourriture ; de telles attitudes sont générées en effectuant différentes associations entre les états internes symboliques (danger, faim, normal) et le but de l'agent. Ils ont élaboré un environnement de simulation, une grille à deux dimensions, où la nourriture est localisée de manière aléatoire. Les agents considérés sont simples et ont besoin de chercher de la nourriture et de manger pour survivre ; ils peuvent interagir avec autrui. Les relations de dépendances entre agents reposent sur la différence entre leur pouvoir (énergie) qui change continuellement en fonction de leurs actions. Quatre types d'agents sont définis. Les **solitaires**, qui ignorent juste l'autre, de sorte à ce qu'il n'y ait pas d'interaction parmi eux ; leur but est toujours de chercher de la nourriture en fonction de leur état interne. Les **parasites** qui, selon leur état interne, restent toujours immobiles, ayant le seul but de chercher de l'aide et de manger de la nourriture reçue d'autres agents ; cela signifie que si les parasites ne sont pas aidés, ils meurent. Les **égoïstes** qui, en fonction de leur état interne, demandent soit de l'aide (lorsqu'ils sont en danger), soit cherchent de manière autonome de la nourriture (lorsqu'ils ont faim ou lorsqu'ils sont dans l'état normal). Le seul comportement social montré par les agents parasites et égoïstes est la recherche d'aide. Enfin, les agents **sociaux** ont différents buts selon leur état interne. En cas de danger, leur générateur de but active le but de chercher de l'aide. Lorsqu'ils ont faim, leur but est de chercher de la nourriture ; lorsqu'ils sont dans l'état normal, s'ils peuvent voir dans leur champ de perception des agents qui recherchent de l'aide, leur but est de donner de l'aide, sinon, ils cherchent de la nourriture. Ils ont donc trois comportements différents selon les variations de leur niveau d'énergie.

Les agents ont un comportement stable durant leur vie qui ne varie pas selon le résultat d'interactions antérieures ou ultérieures avec les autres.

Type d'agent	État interne	But
Solitaire	plusieurs	trouver de la nourriture
Parasite	plusieurs	chercher de la nourriture
Égoïste	Danger Faim, Normal	chercher de l'aide trouver de la nourriture
Social	Danger Normal Faim	chercher de l'aide (si des demandeurs d'aide sont perçus) aider, sinon trouver de la nourriture trouver de la nourriture

Figure 1.3.2.1. : Les diverses attitudes sociales des agents de A. Cesta

L'architecture d'un agent est relativement simple : elle est composée d'un capteur visuel (vue limitée), d'un générateur de but, d'un module de planification, et d'un ensemble d'effecteurs.

L'énergie d'un agent possède deux seuils intermédiaires qui jouent un rôle dans le choix de l'état interne d'un agent. Lorsque le niveau d'énergie d'un agent est compris entre 20 et 60 il a faim, en dessous de 20 il est en danger et au-dessus de 60 il est normal.

Dans les simulations effectuées, les agents choisissent d'aider le récepteur le plus proche et de manger la source de nourriture la plus proche. La quantité de nourriture reste constante pendant toute la durée de la simulation : une fois qu'une source est mangée elle réapparaît aléatoirement dans l'environnement.

Un premier résultat obtenu à partir de comparaisons faites entre les agents sociaux et les exploités (soit les égoïstes, soit les parasites) montre que l'attitude sociale est plutôt robuste. Les agents sociaux, au travers de leur support mutuel, sont capables de compenser l'exploitation dont ils souffrent à cause des agents non réciproques. Cependant, ces observations sont obtenues dans un univers artificiel et en considérant uniquement la variable de survie.

Deux autres résultats ont pu être observés :

a - dans des conditions données, un donneur ne met pas sa vie en danger lorsqu'il décide d'aider autrui puisque l'aide est donnée uniquement lorsqu'il est dans l'état normal. Il est possible de penser que si l'aide était fournie dans des conditions plus risquées, les agents sociaux seraient plus perturbés.

b - Les agents sociaux et les agents égoïstes n'ont pas de différences considérables en terme de survie. Il y a des chances que des différences existent en terme d'accumulation de ressources mais cette caractéristique n'a pas été considérée.

Finalement, pour comparer solitaire égoïste et parasite, ils concluent qu'une attitude d'autosuffisance "modérée", c'est-à-dire, l'autosuffisance combinée avec la recherche d'aide

(égoïste), est plus avantageuse qu'une attitude d'autosuffisance extrême (solitaire). Inversement, cette dernière est plus avantageuse qu'une complète dépendance des autres (parasites) dans le sens où l'aide des autres ne peut pas compenser la totalité des manques d'autosuffisance.

1.3.2.2. Les travaux de Kalenka & Jennings

Suzanne Kalenka et Nick Jennings [Kalenka, 1995] proposent une classification d'attitudes sociales mettant l'accent sur l'engagement implicite des agents dans l'activité collective (leur tâche personnelle) et sur les risques qu'ils prennent lorsqu'ils décident d'aider autrui. Chaque agent est assigné à un rôle particulier dans la collectivité. Trois types distincts d'attitudes peuvent être identifiés. Les agents **responsables** qui exécutent seulement les tâches directement associées à leur rôle dans le groupe. Ils ne doivent pas être confondus avec des agents individualistes : un agent responsable a comme but principal de résoudre ses propres tâches et ne reconsidère jamais cette position. Les agents **serviables** qui assistent les autres lorsqu'ils n'ont plus de tâches personnelles à effectuer dans le temps qui leur est imparti. Ils n'attendent rien en retour. Les agents **coopératifs** qui assistent autrui parce qu'ils supposent que l'aide sera réciproque. Parmi les agents coopératifs, trois sous-classes ont été définies : les agents qui ne prennent pas de risque (ils accordent une plus grande priorité à effectuer leur propre tâche mais éprouvent toutefois une certaine inquiétude pour la collectivité), ceux qui au contraire prennent des risques (ils pensent que le seul moyen d'atteindre leurs propres buts est d'obtenir l'assistance des autres - ils espèrent que leurs propres tâches seront toujours complètement achevées) et les agents altruistes (qui accordent une plus grande priorité aux buts des autres).

La contribution du travail présenté est d'identifier et caractériser les différents types d'engagement qui peuvent survenir durant la résolution d'un problème. Un engagement purement individuel peut être distingué (c'est le cas des agents responsables qui effectuent leur propre tâche). Ensuite l'engagement des agents serviables peut être considéré. Enfin au niveau le plus haut, on distingue les engagements bidirectionnels pris par les agents coopératifs : les agents qui ne prennent pas de risques renoncent à leur engagement lorsque l'atteinte de leur propre tâche est en danger, les agents qui prennent des risques peuvent renoncer à leur engagement lorsque l'atteinte de leur propre tâche est en danger et enfin les agents altruistes ne renoncent jamais à leur engagement.

Les trois attitudes sociales définies et les différents types d'engagement qui en découlent ont été appliqués à la simulation de déchargement de stock dans une usine. Cette application nécessite que les agents coopèrent et coordonnent leur comportement pour résoudre le problème en totalité de manière efficace. Chaque agent est responsable d'une baie dans laquelle doit avoir lieu un déchargement. Deux protocoles de communication ont été définis. Leur invocation est caractérisée

par l'attitude sociale des participants. Un protocole a été défini pour les agents serviables et un autre pour les agents responsables nécessitant de l'aide. Aucun résultat comparatif sur les différents comportements obtenus suite aux attitudes sociales fournies aux agents n'a été explicité.

1.3.2.3. Les travaux de Sekaran & Sen

Sekaran et Sen [Sekaran, 1995] s'intéressent au comportement d'une société composée d'agents plus ou moins coopératifs. Ils définissent 4 types d'agents : les agents **égoïstes** qui demandent de l'aide mais n'aident jamais personne, les agents **individuels** qui ne demandent ni ne fournissent de l'aide à autrui, les agents **philanthropes** qui aident toujours les autres lorsque que cela leur est demandé et les agents **récioproques** qui acceptent ou non (à partir de méthodes stochastiques) une demande de coopération. Ces derniers n'apportent leur aide à autrui que si l'action leur est bénéfique à court ou long terme (ils font le juste poids entre l'énergie supplémentaire que cela leur demande et le bénéfice qu'ils peuvent en retirer par la suite). Dans [Sen, 1996a], Sen insiste sur la différence entre le comportement des agents récioproques et le comportement suivant la règle "un prêté pour un rendu".

Sen attache tout particulièrement de l'intérêt au bénéfice que peut retirer un agent de la coopération. Selon ses propos, un agent ne doit coopérer que s'il en retire un quelconque bénéfice, d'où les agents récioproques. Il argumente sa position en disant que dans un environnement distribué ouvert, l'unique stratégie défendable qu'un agent autonome puisse suivre en décidant ses actions est celui gouverné par son intérêt.

Les différents comportements des agents ont été appliqués au problème de la livraison de paquets. Chacun des N agents doit livrer T paquets contenus dans un dépôt centralisé à une destination définie aléatoirement. Un agent peut seulement transporter un paquet à la fois par lui-même ou avec l'aide d'un autre agent. Lorsqu'il revient au dépôt un nouveau paquet lui est assigné pour livraison. A ce moment là, il regarde si d'autres agents sont dans le dépôt ; si tel est le cas il demande alors de l'aide pour livrer le paquet. Cette demande sera ou non honorée selon les attitudes des agents. L'intérêt d'être aidé est de pouvoir se déplacer plus rapidement donc d'atteindre son but plus rapidement.

Les résultats obtenus montrent, d'une part, que la performance des agents récioproques est quasiment identique à celle des philanthropes. D'autre part, la performance moyenne du groupe repose sur la performance des agents égoïstes et récioproques, et évoluent de manière très proche de la performance des agents égoïstes lorsque le pourcentage de ces derniers est augmenté. Enfin, lorsqu'il y a peu de dépôts à faire, les agents égoïstes peuvent exploiter les agents récioproques pour la plupart de leurs dépôts ce qui leur confèrent momentanément des capacités meilleures.

Leur conclusion est qu'un comportement réciproque peut servir l'intérêt personnel tout autant que l'efficacité globale. Puisque, un comportement réciproque produit, à long terme, une performance meilleure qu'un comportement égoïste ou exploiteur, il est très intéressant pour tous les agents d'être réciproques.

1.4. DISCUSSION

Dans ce chapitre traitant de la coopération, nous ne nous sommes pas intéressés à la coopération au niveau de la résolution de problème distribué, mais au bénéfice de la coopération au niveau de la collectivité. Nous considérons qu'un agent est coopératif, ou plus exactement, pour être cohérent avec ce que nous avons écrit au paragraphe 1.2.1., nous qualifierons le comportement d'un agent de "coopératif" si, compte tenu de ses compétences et de ses croyances, il essaie en permanence de satisfaire autrui. Les compétences et croyances de l'agent évoluent dynamiquement au fur et à mesure de ses interactions grâce à des mécanismes d'apprentissage intégrés dans l'agent (cf. § chapitre 2).

Les différents travaux étudiés, montrent à l'unanimité, que la coopération est bénéfique pour obtenir un comportement proche de l'optimal. Ce résultat n'est pas trivial, car il est possible de croire que l'augmentation du nombre d'agents (non coopératifs certes) dans une société est suffisant pour accroître les performances du système. Cependant, divers travaux ont montré que le caractère individualiste des agents allait à l'encontre de ce phénomène. Dès lors, nous comprenons immédiatement le caractère prépondérant de la coopération.

Faire coopérer les agents en leur attribuant une attitude sociale coopérative nous semble également une bonne méthode. Les agents n'ont alors pas le choix d'avoir un comportement autre que celui imposé par leurs attitudes sociales. Le résultat qui en découle est donc à coup sûr bénéfique pour la collectivité.

Les exemples présentés proposent divers mécanismes pour mettre en oeuvre la coopération. Tag Hogg propose de faire coopérer les agents en leur permettant d'échanger des suggestions. Nagendra Prasad propose qu'ils échangent des points de vue. Dans ces deux cas, la coopération est mise en oeuvre par le biais de la communication indirecte ou directe. Claudia Goldman montre, quant à elle, que la coopération ne nécessite pas forcément de communication et qu'elle peut être issue de l'observation de l'environnement. L'enjeu d'une société où les agents observent leur environnement et sont capables de raisonner sur ces observations, est que les agents puissent s'adapter, et peut-être tirer profit de toute perturbation.

Cela rejoint l'idée que nous allons présenter au niveau de l'apprentissage en univers multi-agent dans le chapitre suivant et qui, au niveau de la coopération, est la suivante : le comportement coopératif sera, à notre sens, plus efficace lorsque les agents seront capables de s'adapter en temps réel aux informations qui proviennent de leur environnement et de leurs congénères. En effet, il serait à la fois ambitieux et restrictif de penser tout prévoir à l'avance. Par conséquent, pour que cette dynamique puisse exister, il est nécessaire que les agents possèdent des attitudes de coopération indépendantes du domaine d'application. Nous adhérons donc totalement à la position de Goldman qui a cherché à associer à ses agents des règles génériques de coopération (méta-règles) qui leur permettent de s'aider les uns les autres implicitement (sous forme de travail supplémentaire).

CHAPITRE 2

L'APPRENTISSAGE DANS LES SYSTÈMES MULTI-AGENTS

“La coordination de multiples agents est essentielle pour la viabilité des systèmes dans lesquels ces agents partagent des ressources. La plupart des recherches en IAD se concentrent sur le développement de stratégies de coordination en différé. Ces stratégies préfabriquées peuvent devenir rapidement inadéquates si le modèle du monde du concepteur du système est incomplet/incorrect ou si l'environnement change dynamiquement. L'apprentissage et l'adaptation sont des mécanismes inestimables grâce auxquels les agents peuvent développer des stratégies de coordination qui satisfont les demandes des environnements et les exigences des agents individuels [Sen, 1996b]”.

2.1. INTRODUCTION

L'apprentissage est un axe de recherche de plus en plus important dans le domaine des systèmes multi-agents car c'est une aptitude nécessaire pour concevoir des systèmes évolutifs et adaptatifs. “L'apprentissage est un aspect intrinsèque de l'intelligence et une nécessité pour s'adapter à un environnement changeant” [Bourret, 1991]. De par le contexte d'utilisation des systèmes multi-agents, l'apprentissage est une faculté indispensable tant pour l'amélioration du fonctionnement du système que pour faciliter le développement d'application. On distingue deux types d'apprentissage :

- ❶ l'apprentissage des compétences, c'est-à-dire des connaissances sur le domaine. L'apprentissage peut s'effectuer de manière isolée ou collective. Dans le premier cas, c'est un agent, qui, de sa propre initiative, crée de nouveaux corpus de connaissances. Dans le second cas, les agents d'un groupe échangent de la connaissance ou s'observent dans leurs activités respectives.
- ❷ l'apprentissage de l'organisation qui porte sur les liens d'interaction possibles entre agents pour leur permettre de mieux se coordonner lors de la résolution d'un problème donné. Il s'agit d'un apprentissage collectif qui consiste “à apprendre les interactions entre agents afin d'améliorer celles-ci” [Weiß, 1996].

A notre connaissance, la grande majorité des travaux porte sur le premier type d'apprentissage, l'apprentissage des compétences, et les méthodes les plus couramment utilisées reposent sur les mécanismes d'apprentissage par renforcement.

Après avoir défini le cadre dans lequel nous nous situons pour étudier les différents mécanismes d'apprentissage, nous expliquerons dans un second paragraphe les principes fondamentaux des algorithmes d'apprentissage par renforcement. Nous étudierons ensuite, dans les deux paragraphes suivants, diverses techniques de mise en oeuvre de l'apprentissage au niveau des compétences puis au niveau de l'organisation. Cette étude sera orientée selon certains critères qui nous semblent capitaux pour la mise en oeuvre de l'apprentissage dans un contexte multi-agent. Ces trois caractéristiques seront explicitées et justifiées à la fin du paragraphe suivant.

2.2. CADRE DE L'ÉTUDE

Pour faire face aux éventuelles perturbations de l'environnement dans lequel il est plongé, un système multi-agent doit être capable de s'adapter à toutes les situations. "Les capacités d'adaptation et d'auto-organisation d'un système est fortement liée à sa faculté d'apprendre avec lui-même ainsi qu'avec le monde qui l'entoure" [Anacleto, 1996]. Un agent s'adapte au sein d'un système multi-agent, si, face à une situation imprévue par le concepteur de l'application finale, l'agent ne se bloque pas mais réagit. Lorsqu'un agent s'adapte à une perturbation, il abandonne alors ponctuellement, voire parfois définitivement, l'objectif individuel qu'il poursuit. L'agent modifie alors son comportement face aux évolutions de son environnement sans en tirer de conséquences durables. Ainsi, pour deux situations identiques rencontrées au cours de son histoire, l'agent adaptatif se comportera exactement de manière identique.

Trois conditions de fonctionnement d'un agent face à des situations imprévues peuvent être observées.

- ① Le concepteur du système multi-agent et le concepteur de l'application ne donnent pas les capacités à l'agent de détecter et de traiter ces situations. Dans ces conditions, l'agent essaye d'atteindre son objectif. Si une situation imprévue survient, il agit sans avoir détecté ce type de situation, donc de manière moins pertinente du point de vue du fonctionnement du système. Il peut être en particulier gêné pour faire face à cette situation ou peut bloquer involontairement l'activité d'autres agents.
- ② Le concepteur du système multi-agent donne à l'agent des outils pour traiter les situations imprévues et le concepteur de l'application utilise ces mécanismes pour donner à l'agent la capacité de détecter certaines situations et d'y réagir. Cependant, toutes les situations inattendues ne peuvent pas être exhaustivement prévues par le concepteur de l'application car

elles surviennent comme des perturbations par rapport au fonctionnement pour lequel le système a été conçu. Les situations non définies *a priori* seront traitées comme dans le cas ①.

- ③ Le concepteur du système multi-agent dote l'agent d'attitudes sociales qui lui permettent de détecter et de réagir aux situations imprévues. Il doit pour cela instancier les connaissances génériques du comportement de l'agent. Les agents peuvent ainsi traiter des situations inattendues sans les connaître toutes *a priori*. Au moment où les agents détectent une situation imprévue (gênante ou non pour leur résolution), ils renoncent à atteindre leur objectif courant pour supprimer ce type de situation. C'est en cela qu'ils sont adaptatifs. Quand la situation imprévue a été traitée, ils peuvent soit poursuivre l'objectif momentanément abandonné, soit en changer.

Nous nous plaçons dans ce dernier cas pour étudier le mécanisme d'apprentissage dont nous souhaitons doter nos agents. Tout en s'adaptant de la sorte, les agents doivent, selon nous (cf. partie 2), apprendre en s'auto-organisant dès qu'une situation imprévue survient. Ainsi, lorsque deux situations imprévues identiques se présenteront, le système n'aura pas forcément le même comportement. Ce comportement sera fonction des connaissances actuelles qui auront sûrement évolué depuis la première situation. L'apprentissage qui nous intéresse porte exclusivement sur l'organisation. Comme l'apprentissage de l'organisation est une résultante des échanges d'un agent avec son environnement, nous étudierons également trois méthodes d'apprentissage sur les connaissances du domaine, ces méthodes reposant sur l'interaction. Nous présenterons enfin cinq méthodes d'apprentissage sur l'organisation en univers multi-agent.

Ce chapitre n'est pas un état de l'art exhaustif des techniques d'apprentissage dans les systèmes multi-agents car nous focalisons notre exposé sur les méthodes les plus pertinentes pour notre objet d'étude.

Trois caractéristiques nous paraissent importantes pour la mise en oeuvre de l'apprentissage en univers multi-agent [Camps, 1996b]. En effet, les systèmes multi-agents ont des caractéristiques essentielles qui nous semblent contraindre la mise en oeuvre de l'apprentissage. Il s'agit de :

- la **distribution** : les systèmes multi-agents sont caractérisés par la distribution à la fois des données et du contrôle. Il paraît donc logique de distribuer le mécanisme d'apprentissage dans chaque agent qui apprendra alors à partir des informations locales qu'il connaît.
- l'**asynchronisme** : les agents dans les systèmes multi-agents sont autonomes. Cette autonomie leur permet d'avoir un fonctionnement asynchrone au niveau de la communication, du raisonnement... Pour conserver cette notion d'autonomie au niveau de l'apprentissage, il faut que l'agent puisse décider en fonction de ce qu'il perçoit de l'instant où il doit apprendre.
- la **coopération** : le problème essentiel introduit par la distribution est qu'aucun agent ne possède une vue complète des connaissances. C'est pourquoi selon nous, les agents doivent coopérer.

Grâce à la coopération, les agents arriveront à faire plus de choses que s'ils avaient travaillé seuls, de manière individuelle.

Outre ces trois propriétés que doit posséder notre algorithme d'apprentissage, nous désirons également que l'apprentissage s'effectue sans faire de présupposé sur le résultat qu'est censé fournir le système (cf. §2 dans introduction).

Avant de passer à l'étude des diverses méthodes d'apprentissage, nous allons expliciter les mécanismes les plus utilisés lors de la mise en oeuvre de l'apprentissage par renforcement.

2.3. L'APPRENTISSAGE PAR RENFORCEMENT

Selon Singh [SinghS, 1996], la résolution de problèmes complexes reposant sur des techniques décrivant les actions à effectuer dans un état de l'environnement donné est très coûteuse. Il considère que de tels problèmes requièrent une nouvelle manière de penser la résolution et préconise les algorithmes par renforcement à cet effet. Ils permettent, selon lui, une performance meilleure et plus robuste dans un monde réel changeant et dans les environnements incertains. Ils permettent en outre aux agents d'apprendre des solutions voisines de l'optimal tout en utilisant des spécifications de haut niveau.

Trois méthodes d'apprentissage par renforcement utilisent à des degrés divers l'appréciation (feed-back ou rétroaction) de l'environnement ou des agents constituant le système. On distingue :

- l'apprentissage supervisé dans lequel le feed-back de l'environnement spécifie précisément l'information sur la sortie désirée. L'objectif de l'apprentissage est d'arriver à se rapprocher de cette information le plus étroitement possible,
- l'apprentissage par renforcement dans lequel le feed-back spécifie seulement l'utilité de l'activité actuelle de l'apprenant. L'objectif est alors de maximiser cette utilité,
- l'apprentissage non supervisé dans lequel il n'y a pas de feed-back explicite fourni. L'objectif est alors de trouver des activités utiles et désirées en se basant sur des processus d'essai/erreur ou d'auto-organisation.

L'environnement ou l'agent fournissant le feed-back fait donc office de "maître" dans le cas d'un apprentissage supervisé, de "critique" dans le cas d'un apprentissage par renforcement et enfin d'observateur passif dans le cas d'un apprentissage non supervisé.

L'apprentissage supervisé est particulièrement utilisé dans les réseaux connexionnistes. L'algorithme de rétro propagation en est un exemple typique : il consiste à présenter simultanément

au réseau une entrée et la sortie désirée de façon à pouvoir calculer la différence entre la sortie produite et la sortie désirée et minimiser l'erreur commise en changeant la pondération des liens entre neurones. Ce type d'apprentissage ne correspond pas aux hypothèses que nous avons émises car nous souhaitons une méthode d'adaptation suffisamment générique pour ne pas faire de présupposé sur la finalité du résultat escompté. Étant donné que l'apprentissage correspond à la mémorisation d'une action d'adaptation suite à une perturbation particulière, il ne doit pas être conditionné par une sortie attendue. L'apprentissage ne peut donc en aucun cas être supervisé.

Dans un apprentissage par renforcement, l'exécution d'une action produisant des résultats satisfaisants est renforcée donc récompensée et est affaiblie dans le cas contraire. Chaque agent, à partir d'une description de l'état courant, doit choisir la prochaine action à exécuter de façon à maximiser le retour qu'il recevra une fois l'action effectuée. L'apprentissage est fondé sur une corrélation entre les comportements effectués et les résultats qu'il reçoit en retour. Les algorithmes d'apprentissage par renforcement tentent de maximiser les récompenses et de minimiser les punitions au cours du temps.

Ce type d'apprentissage étant très usité dans les articles sur lesquels nous nous basons pour cet état de l'art, nous allons présenter les trois algorithmes les plus référencés : il s'agit des algorithmes du Q-learning, du "bucket brigade" et des systèmes classifieurs.

Enfin, l'apprentissage non supervisé permet d'apprendre de manière totalement autonome et indépendamment d'un quelconque contrôle. Ce mode d'apprentissage est idéal dans le cas de systèmes auto-organiseurs. Ces derniers peuvent alors s'adapter et apprendre en fonction des perturbations en provenance de l'environnement. Ce type d'apprentissage nous intéresse tout particulièrement mais il est très peu utilisé dans la littérature.

2.3.1. L'algorithme du Q-learning

L'algorithme du Q-learning [Watkins, 1992], [Mataric, 1994] est l'algorithme le plus connu parmi les techniques d'apprentissage par renforcement. Son fonctionnement est le suivant : il travaille en estimant les valeurs de paires (état, action) - (s,a) - et essaie de maximiser la valeur $Q(s, a)$. La valeur $Q(s,a)$ est la récompense discontinue attendue lorsque l'action a est effectuée dans l'état s. Les valeurs Q pour toutes les paires état-action sont sauvegardées dans une table et sont mises à jour à chaque étape. L'utilité E d'un état est la valeur Q maximum de toutes les actions qui peuvent être prises dans cet état. La valeur Q pour effectuer une action dans un état est définie comme la somme de la récompense r immédiate et de l'utilité $E(s')$ de l'état suivant s' selon la

fonction de transition d'état T , prévue par le paramètre γ . Formellement [Mataric, 1994], cela peut s'écrire : $s' \leftarrow T(s,a)$, $E(s) = \max_a Q(s, a)$, $Q(s,a) = r + \gamma E(s')$, $0 \leq \gamma \leq 1$.

La mise à jour de $Q(s, a)$ s'effectue selon la formule ci-dessous :

$$Q(s, a) \leftarrow Q(s, a) + \beta[r + \gamma E(s') - Q(s, a)] \text{ avec } 0 \leq \beta \leq 1, \text{ où } E(s') = \max_b Q(s', b)$$

β détermine l'estimation de l'apprentissage. Lorsque $\beta = 1$ l'apprentissage ne tient pas compte de tout l'historique accumulé dans la valeur Q courante et Q est réinitialisée avec la somme des récompenses reçues et attendues à chaque étape. γ est le facteur discontinu pour la récompense future. Idéalement γ devrait être aussi proche que possible de 1 de sorte que la récompense future soit maximisée.

2.3.2. Les systèmes classifieurs

Les systèmes classifieurs [Holland, 1985] sont couplés à des algorithmes par renforcement. Ces systèmes, constitués de règles, apprennent en ajustant la force associée à leurs règles à partir d'un feed-back et en découvrant de meilleures règles en utilisant un algorithme génétique. Ils sont essentiellement utilisés pour créer des comportements adaptatifs.

De manière plus précise, les systèmes à base de classifieurs sont très proches des règles de productions. Les règles, appelées classifieurs, sont des couples de chaînes de caractères dont les éléments appartiennent à l'alphabet $\{0, 1, \#\}$ (le caractère '#' ne possède pas de valeur mais peut prendre la valeur 0 ou 1). Un poids est associé à chaque règle et reflète la probabilité qu'elle soit déclenchée. Les faits, appelés messages, sont des chaînes de caractères constituées de caractères appartenant à l'alphabet $\{0, 1\}$. Elles sont de taille fixe. Les informations en provenance de l'environnement sont codées sous la forme d'un ensemble de faits qui peuvent déclencher des règles ou des actions dans l'environnement. A ce mécanisme, similaire aux systèmes de production, s'ajoute un système de récompense (bucket brigade, ...) qui gratifie les règles qui ont entraîné l'exécution d'une action ayant permis de faire avancer la résolution du problème courant, mais pénalise, au contraire, les règles n'ayant pas contribué à la résolution. Les règles ainsi pondérées et dont le poids est le plus élevé sont déclenchées en priorité. Outre ce mécanisme d'attribution, un classifieur possède un algorithme génétique permettant, à l'aide des opérateurs de mutation et de croisement (cf. §3.7) de reproduire de nouvelles règles à partir des règles ayant le plus grand poids.

Voici l'exécution du cycle de base d'un classifieur présentée dans [Shaw, 1989] :

- ① Placer tous les messages en provenance de l'interface en entrée dans la liste courante des messages
- ② Comparer tous les messages à toutes les conditions des règles et former un ensemble M contenant les règles qui ont leur partie condition satisfaite
- ③ Calculer la récompense de chaque règle dans M
- ④ Sélectionner la règle R ayant la plus haute offre b(r)
- ⑤ Réduire la force de R de b(r)
- ⑥ Déterminer l'ensemble des règles qui ont envoyé des messages qui s'apparient avec les conditions de R et augmenter leur force de $b(r) / |M|$ où $|M|$ est le nombre de règles dans M
- ⑦ Générer un nouveau message à partir de la partie action de R pour la liste de messages
- ⑧ Exécuter la nouvelle liste de messages vers l'interface de sortie pour produire la sortie du système
- ⑨ Si une récompense est reçue en provenance de l'environnement, l'allouer aux règles qui ont participé à la génération de messages
- ⑩ Revenir à l'étape 1

Figure 2.3.2. : Cycle de base d'un classifieur [Shaw, 1989]

2.3.3. L'algorithme du "bucket brigade"

L'algorithme du "bucket brigade" [Holland, 1985] est un schéma d'apprentissage par renforcement très souvent couplé aux classifieurs. Il fonctionne de la manière suivante : lorsqu'un agent est choisi pour être exécuté, sa force est augmentée par le feed-back environnemental. Mais avant cela, une fraction α de sa force est ôtée et ajoutée à la force du classifieur qui a été exécuté dans le dernier cycle.

Ainsi, si un classifieur i, exécuté à l'étape t, produit un feed-back externe de R, et le classifieur j s'exécute à la prochaine étape, l'équation suivante [Sen, 1995] donne la force mise à jour du classifieur i.

$$S_{t+1}(c_i, a_i) = (1 - \alpha) * S_t(c_i, a_i) + R + \alpha * S_{t+1}(c_j, a_j)$$

2.4. L'APPRENTISSAGE DES CONNAISSANCES DU DOMAINE

L'apprentissage des connaissances pour résoudre un problème concerne soit l'acquisition de nouvelles connaissances sur le domaine soit l'assimilation de meilleures stratégies de résolution. Plusieurs auteurs se sont penchés sur l'étude de l'apprentissage des compétences : on distingue notamment Sian [Sian, 1991], Mataric [Mataric, 1996], Sen and Sekaran [Sen, 1994]. Chaque agent apprend sur le domaine d'application pour résoudre les problèmes de manière plus efficace.

Bien que l'on se situe dans le cadre de l'intelligence artificielle distribuée, les agents ont la possibilité d'apprendre isolément ou en groupe.

Un apprentissage groupé permet :

- une meilleure coordination qui peut être réalisée grâce aux partages d'informations, ou à un meilleur signalement entre les agents,

- une meilleure efficacité de la tâche et de l'allocation de ressources. Cette amélioration peut être expliquée par la spécialisation des agents pour certaines tâches : le groupe peut par exemple apprendre soit certaines de ses caractéristiques (les agents a et b travaillent mieux ensemble), le modèle des tâches (pour un type de problème, il est plus facile de scinder le problème en deux, les tâches a et b, et commencer par b), les caractéristiques de l'environnement (comme les préférences de l'utilisateur ...).

Dans le cas d'un apprentissage individuel, chaque agent peut apprendre de son côté en ajustant ses vues ou ses actions. On distingue également deux types d'activités d'apprentissage :

- un agent peut apprendre pour améliorer sa résolution en utilisant des heuristiques, la révision des croyances et l'utilisation de connaissances plus pertinentes,

- en observant comment les autres agents résolvent les problèmes. Avec la présence d'autres agents dans le système, les capacités d'un agent à résoudre un problème peuvent être améliorées. Cette amélioration peut se faire en intégrant l'expérience d'autres agents à partir d'une explication [Piquemal, 1994], ou tout simplement en demandant de l'aide pour résoudre un type de problème particulier.

L'apprentissage en groupe nous semble plus adéquat au domaine d'application : les agents, au fur et à mesure de leurs interactions, peuvent apprendre sur les autres et mettre à jour leurs compétences en conséquence. Nous ne parlerons donc pas, par conséquent, d'apprentissage individuel.

2.4.1. Méthodes d'apprentissage sur les compétences

Trois méthodes distinctes permettant l'apprentissage des connaissances du domaine vont être présentées dans ce paragraphe. Le premier article (Sen) fait abstraction de l'interaction bien que cela soit à notre sens nécessaire dans un contexte multi-agent. Le second article montre que si les agents coopéraient, les résultats obtenus seraient meilleurs. Enfin le dernier article se base sur la coopération pour apprendre.

2.4.1.1. Le modèle de Sekaran & Sen

Le modèle présenté dans [Sekaran, 1994] utilise l'algorithme du Q-learning. Après s'être intéressés à l'apprentissage dans un domaine coopératif où deux agents travaillent ensemble sur une tâche commune sans partage explicite de connaissances ou d'informations [Sen, 1994], Mahendra Sekaran et Sandip Sen étendent leur approche à un domaine non coopératif, où les agents ont des buts conflictuels. Ces agents ne partagent ni connaissance, ni résultat. Ils apprennent une stratégie à partir d'observations et l'améliorent avec l'expérience. L'originalité de ce travail est qu'il n'y a ni échange de connaissance explicite entre les agents ni dépendance de relations.

Cette méthode est appliquée au monde des blocs : deux agents autonomes, ayant différentes capacités et aucune connaissance sur le domaine, doivent pousser un bloc d'une position initiale à une position finale dans un espace euclidien. A chaque étape, chaque agent applique une force suivant un certain angle et le bloc est déplacé. La session est terminée dès que le bloc atteint une des deux positions finales ou lorsque le bloc sort du champ de jeu. Chaque agent apprend, par l'intermédiaire de tentatives répétées et en interagissant, une politique optimale pour atteindre la position finale convoitée. Le retour (feed-back) reçu après chaque action est une fonction inversement exponentielle de la distance entre la position courante du bloc et le chemin optimal associé à leurs buts respectifs. Les agents mettent alors à jour leur politique individuelle en utilisant l'algorithme du Q-learning et en fonction du feed-back reçu. La convergence des matrices de stratégies (il n'y a pas de mise à jour significative des politiques individuelles sur un ensemble d'essais) est utilisée comme critère d'arrêt pour les expériences.

Les auteurs montrent que les agents peuvent, pour atteindre un but particulier, utiliser des schémas d'apprentissage de renforcement pour atteindre leur but, qu'ils soient coopératifs ou conflictuels, et sans avoir de modèle l'un de l'autre.

2.4.1.2. Le système de Mataric

Maja Mataric [Mataric, 1996], [Mataric, 1994] trouve les algorithmes d'apprentissage par renforcement inadaptés à la problématique des robots pour deux raisons : les robots ont parfois plusieurs buts à atteindre et chaque robot possède un espace d'états très vaste. Selon lui, puisque les agents ont de multiples buts, il est plus simple de renforcer chacun d'eux individuellement avec une fonction de renforcement hétérogène plutôt que d'essayer de les comprimer par une unique fonction but. Cependant, les buts multiples ne sont pas suffisants pour accélérer l'apprentissage situé en particulier lorsque chaque but nécessite une séquence complexe d'actions. Mataric propose

pour cela des estimateurs de progrès c'est-à-dire des fonctions qui permettent de fournir un renforcement positif ou négatif basé sur le progrès immédiat mesuré par rapport aux buts spécifiques.

Les expériences effectuées portent sur un groupe de quatre robots fourrageurs totalement autonomes. Chaque robot apprend individuellement à sélectionner le meilleur comportement pour chaque condition, de sorte à trouver et ramener le plus de palets dans sa maison. A cet effet, quatre comportements et quatre variables d'états ont été définis. Certains événements particuliers produisent des renforcements positifs immédiats "saisir un palet", "déposer un palet à la maison" et "se reposer à la maison" alors que d'autres entraînent un renforcement négatif immédiat "déposer un palet hors de la maison", "se reposer hors de la maison". Tous sont combinés dans une fonction de renforcement hétérogène $E(c)$. Deux fonctions d'estimation de progrès sont également utilisées :

- la première, I , est déclenchée lorsqu'un agent est très proche d'un autre. Son but est d'éviter les interférences entre robots. Si le comportement exécuté a pour effet d'augmenter (resp. diminuer) la distance physique avec l'autre agent, l'agent reçoit un renforcement positif (resp. négatif). Formellement $I(c,t) = m$ si la distance entre les deux agents diminue, n sinon avec $m > 0$ et $n < 0$.

- la seconde, H , concerne le retour à la maison et est initiée lorsqu'un palet est saisi. Si la distance jusqu'à la maison décroît (resp. croît) pendant que H est actif, l'agent reçoit un renforcement positif (resp. négatif). Un statu quo ne délivre pas de renforcement. Formellement $H(c,t) = i$ si l'agent est plus près de la maison, j s'il est plus loin de la maison, 0 sinon, avec $i > 0$ et $j < 0$.

L'apprentissage est continu et incrémental durant la vie de l'agent, assurant ainsi que l'agent reste responsable des changements dans l'environnement (il ne reste plus de palets dans une zone particulière) et des changements internes (panne d'un capteur ...).

Mataric teste trois fonctions différentes permettant d'effectuer un apprentissage par renforcement :

- une fonction de renforcement pour un but unique monolithique (un palet à porter dans la région de la maison) en utilisant le Q-learning $R(t)=P(t)$
- une fonction de renforcement hétérogène utilisant des buts multiples $R(t)=E(t)$
- une fonction de renforcement hétérogène utilisant des buts multiples et les deux fonctions d'estimation de progrès $R(t)=E(t)+I(t)+H(t)$

Les résultats montrent que l'algorithme du Q-learning est très vulnérable aux interférences des autres robots et que ses résultats ont vite décliné (comparativement aux trois approches) lorsqu'il a été testé sur des groupes de grande taille. Le second algorithme fournit de meilleurs résultats que le Q-learning car il détecte l'atteinte de sous-buts. Enfin, le troisième algorithme maximise l'utilisation de toute l'information disponible dans chaque condition et comportement.

2.4.1.3. Le modèle de Sian

L'originalité de la méthode d'apprentissage proposée par Sati Singh Sian [Sian, 1991] repose sur l'utilisation de la coopération comme mécanisme d'adaptation et d'apprentissage. Elle s'applique à des sociétés d'agents autonomes dans lesquelles aucun agent ne possède une information complète. Ils doivent alors coopérer dans le but de posséder une information plus complète et par là même bénéficier de l'expérience (positive ou négative) d'autres agents. Sian fait le postulat suivant : comme tous les agents opèrent dans un environnement similaire, ils peuvent attendre de l'information utile en provenance des autres agents.

Le mécanisme d'apprentissage proposé s'effectue en trois phases :

- la génération d'hypothèses initiales par un des agents. Des mécanismes ont été définis pour spécifier le moment où doit s'effectuer cette génération. A chaque hypothèse est associée une valeur de confiance basée sur le nombre d'instances utilisées pour la générer.
- la coopération des agents pour arriver à une version d'hypothèses acceptable par tous. Sa mise en oeuvre nécessite une représentation des connaissances homogènes, la définition de fonctions d'évaluation d'hypothèses et un mécanisme d'unification des diverses opinions.
- l'intégration des informations apprises par les agents. L'apprentissage a lieu quand tous les agents ont réussi à trouver un consensus sur l'hypothèse considérée. A ce niveau, le problème de la révision des croyances se pose pour intégrer les nouvelles connaissances. La révision est non monotone.

Les agents communiquent au moyen d'un tableau noir et d'un langage composé de 9 opérateurs relatifs à la gestion d'hypothèses comme, par exemple, changer le statut d'une hypothèse, n'exprimer aucune opinion sur une hypothèse...

Tous les agents sont dotés d'un mécanisme d'apprentissage identique. Un agent décide du choix des hypothèses sur lesquelles va porter l'apprentissage et sollicite les autres pour apprendre. Ainsi le mécanisme d'apprentissage est distribué et garantit l'autonomie des agents. Les agents confirment ou infirment, par des interactions coopératives, des hypothèses qu'ils ont déduites. Ils utilisent pour cela un mécanisme d'induction sur des exemples d'événements ayant déjà eu lieu auparavant.

Le modèle présenté, appelé MALE (Multi-Agent Learning Environment) et écrit en PROLOG, a été appliqué au marché des matières premières. Le système, constitué de trois agents ayant des connaissances concernant différents pays (l'Afrique, l'Amérique et l'Asie) tente de prédire les prix des matières premières en fonction de différents facteurs. La coopération permet à chaque agent de bénéficier de l'expérience des autres en plus de la sienne. Le contenu des messages échangés est alors bien plus intéressant que dans le cas d'agents travaillant individuellement.

2.4.2. La distribution des algorithmes

Nous allons dans ce paragraphe voir si les mécanismes d'apprentissage proposés par chacun des auteurs précédents sont distribués ou non dans chaque agent.

Le mécanisme d'apprentissage proposé par Sian est distribué dans chaque agent. Il permet d'effectuer principalement deux tâches : décider du statut d'une nouvelle hypothèse et permettre l'apprentissage coordonné via une structure de communication commune, un tableau noir. La fonction permettant à chaque agent d'associer une force à une croyance est, quant à elle, globalement connue pour permettre une comparaison significative.

Les agents de Sen et Sekaran, quant à eux, ne se connaissent pas les uns les autres et ne partagent aucune connaissance. L'algorithme d'apprentissage proposé est basé sur le Q-learning et est distribué dans chaque agent.

Le mécanisme d'apprentissage proposé par Mataric est identique et distribué dans chacun des robots.

2.4.3. L'asynchronisme des algorithmes

Nous allons à présent voir si dans les méthodes d'apprentissage proposées, les agents ont besoin de se synchroniser pour apprendre ou si chaque agent peut décider en fonction de ce qu'il sait du moment où il doit apprendre.

Les agents de Sian jugent de la pertinence d'une hypothèse en échangeant leurs opinions par le biais d'un tableau noir. L'apprentissage a donc lieu quand les agents ont réussi à trouver un consensus sur l'hypothèse considérée. Mais cependant chaque agent ne donne son avis sur l'hypothèse en cours que lorsqu'il le désire. Tous les agents ne doivent pas se synchroniser pour donner leur avis et pour mettre à jour leurs croyances (ils pourront mettre à jour leurs croyances quand ils le veulent du moment que le consensus a eu lieu). Nous pouvons donc dire, de ce point de vue là, que l'apprentissage est asynchrone.

Les agents de Sen et Sekaran apprennent en fonction du feed-back qu'ils reçoivent en provenance de leur environnement une fois un déplacement effectué. Ils apprennent de manière totalement

indépendante puisqu'ils n'ont pas de connaissance les uns sur les autres. Le mécanisme d'apprentissage est donc asynchrone.

Tous les robots de Mataric font ce qu'ils veulent quand ils veulent. Ils sont totalement autonomes et apprennent par conséquent de manière totalement asynchrone.

2.4.4. La coopération dans les algorithmes

Nous allons dans ce paragraphe voir si les mécanismes d'apprentissage proposés par chacun des auteurs précédents reposent sur la coopération pour apprendre.

Dans le modèle de Sian, la coopération est le moteur de l'apprentissage : les agents coopèrent pour évaluer la pertinence d'une hypothèse générée et, en fonction de la pertinence qui lui est attribuée, l'apprennent ou ne l'apprennent pas.

Sen et Sekaran montrent qu'une coordination globale entre les agents peut émerger bien que les agents ne partagent pas de connaissance de résolution de problème, donc agissent indépendamment et de manière autonome, c'est-à-dire apprennent en optimisant indépendamment leur propre réponse environnementale. Sen et Sekaran qualifient leurs agents de conflictuels alors que, selon la définition que nous donnons au paragraphe 5.2, les agents sont à notre sens concurrents. Ils ont un but respectif qui est identique à celui de l'autre. La coordination entre eux s'effectue par le biais de l'environnement. Les travaux de Sen et Sekaran s'orientent à l'heure actuelle vers les agents réciproques ; ce sont des agents acceptant ou non de coopérer selon le bénéfice à court ou long terme qu'ils peuvent tirer de l'action [Sekaran, 1995]. L'aspect coopératif leur semble donc intéressant dans la mise en oeuvre d'un mécanisme d'apprentissage puisqu'ils essaient de "promouvoir des comportements coopératifs" [Sen, 1996a].

Enfin Mataric s'est rendu compte et l'a écrit dans sa thèse [Mataric, 1994] que lorsque le nombre de robots augmente, il y a un accroissement des interférences entre les agents et donc un déclin de performance pour les algorithmes par renforcement classiques. Pourtant dans un scénario idéal, les autres agents devraient favoriser l'apprentissage individuel : ce serait effectivement le cas si les algorithmes étudiés reposaient sur un apprentissage mutuel donc si les agents coopéraient.

Dans les méthodes proposées par Sen & Sekaran et Mataric, les agents n'interagissent certes pas entre eux, mais interagissent avec leur environnement physique. Chaque agent communique avec son environnement physique en agissant sur lui. Son action transforme l'environnement et amène

ce dernier à interagir avec l'agent pour lui signaler entre autres son état par rapport à l'état recherché. Ces rétroactions de l'environnement permettent à l'agent, en fonction de son algorithme d'apprentissage, de guider son comportement.

2.5. L'APPRENTISSAGE DE L'ORGANISATION

Dans cette forme d'apprentissage, les interactions sont étudiées pour améliorer la capacité de tout le système. Cela permet une meilleure coordination entre agents et une meilleure allocation de tâches et de ressources. La solution est alors atteinte plus rapidement et de manière moins coûteuse (avec moins de ressources).

La plupart des articles traitant de ce mode d'apprentissage utilisent les algorithmes comme l'algorithme du Q-learning [Weiß, 1993] ou les systèmes classifieurs [Shaw, 1989].

2.5.1. Méthodes d'apprentissage sur l'organisation

Les cinq méthodes que nous allons présenter concernent l'apprentissage de l'organisation à des degrés divers. Shaw propose une méthode permettant à la fois d'apprendre les connaissances sur le domaine et sur l'organisation (création de nouveaux agents à l'aide d'algorithmes génétiques). Weiß propose deux algorithmes d'apprentissage, issus de l'algorithme du "bucket brigade", visant à une meilleure coordination entre les agents de la société. Foisel s'intéresse à la gestion des liens d'interaction entre agent ; chaque agent calcule l'intérêt qu'il a à participer à une organisation en évaluant le gain qu'il retire de chacune de ses interactions. Prasad propose une méthode d'apprentissage permettant aux agents d'apprendre le rôle organisationnel qu'ils doivent tenir dans une recherche négociée pour obtenir des modèles acceptables. Enfin Terabe s'intéresse à l'établissement de liens d'interaction entre agents permettant d'exécuter des tâches en un minimum de temps (en minimisant notamment les communications entre agents).

2.5.1.1. La méthode de Shaw & Whinston

La méthode de Shaw [Shaw, 1989] est destinée à des sociétés d'agents cognitifs, qui communiquent par envoi de messages. Le mécanisme d'apprentissage qu'il propose est basé sur deux processus : un processus d'offre de résolution de tâches et un processus de transformation génétique.

Le processus d'offre est une extension du réseau contractuel de Smith et Davis [Smith, 1980] dans lequel les agents travaillent par partage de tâches en respectant le cycle suivant : décomposition d'une tâche en sous-tâches, allocation des sous-tâches aux agents compétents, résolution des sous-tâches par les agents choisis et intégration des solutions partielles. L'extension consiste à récompenser les agents sélectionnés pour résoudre une tâche au cours du processus de résolution. Cette récompense permet d'augmenter la "force" d'un agent. Cette force quantifie le service rendu par l'agent à la collectivité, elle reflète sa capacité à résoudre des tâches. Elle est utilisée ensuite pour déterminer quelles sont les caractéristiques importantes des agents.

Grâce au processus de détection des propriétés désirables, des algorithmes génétiques de mutation ou de croisement sont utilisés pour remplacer les agents "faibles" par des agents "forts". La performance globale du système est ainsi augmentée. Pour ce faire, chaque agent possède une représentation génétique de ses caractéristiques et de ses capacités. Les agents qui interviennent le plus dans le processus de résolution sont choisis pour avoir leurs capacités dupliquées car ils possèdent les forces les plus grandes. La coopération est le critère utilisé pour guider l'apprentissage au sein de ce système. Shaw l'exprime ainsi : "The better agents that have been successfully completing more tasks would increasingly be favored by the bidding process..., the less useful agents are forced to adapt and change their configuration by mimicking some of the features of the successful agents."

L'architecture des systèmes à base de classifieurs a été appliquée de manière distribuée au sein de chacun des agents pour permettre l'apprentissage. Par contre, les algorithmes génétiques sont centralisés car ils nécessitent la connaissance des forces assignées à tous les agents du système.

Cette méthode a été mise en oeuvre dans le cadre d'un système de fabrication. Chaque agent est un intervenant au niveau de la fabrication et les chromosomes représentent tout ce que l'agent sait faire. Dans ce système, l'apprentissage concerne à la fois les connaissances des agents (puisque les agents qui vont être modifiés intègrent les caractéristiques des agents les plus sollicités) et à la fois l'organisation (puisque cette modification entraîne une réorganisation du système par le biais d'interactions futures).

2.5.1.2. La méthode de Weiß

Weiß [Weiß, 1993] propose une méthode destinée à des sociétés d'agents réactifs qui communiquent constamment avec leur environnement et entre eux par envoi de messages. Cette méthode s'inscrit dans le cadre d'un apprentissage collectif. Les agents possèdent une vision incomplète de leur environnement. Un agent peut donc posséder une information locale sur l'état de l'environnement qui diffère de celle que possède un autre agent. Il n'y a pas de hiérarchie entre les agents.

La méthode d'apprentissage distribuée préconisée vise une meilleure coordination entre les agents de la société. Pour cela, deux algorithmes d'apprentissage, ACE (Action Estimation) et AGE (Action Group Estimation), tous deux issus de l'algorithme du "bucket brigade", ont été développés. Le traitement réalisé par ces deux algorithmes peut être décrit de manière simplifiée comme l'exécution répétée du cycle suivant :

- la détermination de l'action. Dans une première étape chaque agent détermine, suivant ce qu'il connaît de l'état de l'environnement, l'ensemble des actions qu'il peut effectuer ;
- la compétition. Les agents concourent pour le droit de devenir actif. Cette compétition englobe le calcul et l'annonce de l'offre ainsi que la sélection des actions qui sont actuellement effectuées ;
- la répartition des crédits. Dans cette dernière étape, les agents attribuent à l'aide de la méthode du "bucket brigade", un crédit à chacun des autres agents selon l'estimation de leurs actions.

Les deux algorithmes ACE et AGE proposés se différencient par leur manière de mettre en oeuvre la compétition.

Dans la méthode ACE, les agents concourent pour pouvoir effectuer des actions individuelles. Chaque agent fait une offre pour chacune des actions qu'il peut effectuer et l'annonce à tous les autres agents qui sélectionnent alors l'offre la plus élevée parmi celles proposées. L'agent qui possède cette dernière est autorisé à exécuter l'action choisie. Tous les agents suppriment ensuite toutes les offres associées à des actions incompatibles avec l'action sélectionnée. Ce traitement est réitéré jusqu'à ce qu'il n'y ait plus d'action ayant une offre non nulle à sélectionner.

La méthode AGE est très proche de la méthode ACE si ce n'est que les agents concourent pour effectuer des groupes d'actions et non plus des actions individuelles. Pour cela, un agent estime la pertinence d'un but ou d'une action non seulement sur la dépendance entre sa connaissance et l'état courant de l'environnement mais aussi sur la dépendance des contextes des activités possibles. Une telle séquence nécessite la connaissance des autres et la possession d'un même algorithme d'apprentissage notamment pour le calcul des offres. Il est essentiel que les agents soient rationnels et non égoïstes ; il ne faut pas que les agents insistent pour exécuter une action incompatible avec les actions précédemment exécutées. Le choix d'une offre parmi toutes celles proposées induit également un contrôle centralisé ce qui supprime l'autonomie de l'agent pour l'apprentissage.

Ces deux algorithmes ont été étudiés expérimentalement dans le monde des blocs. Les agents doivent à partir d'une configuration initiale aboutir à une configuration finale. Chaque agent est responsable d'un cube. Tout en ayant une vue limitée de l'état de l'environnement, les agents parviennent à agencer les cubes pour atteindre la configuration désirée. Cette application montre que l'algorithme AGE fournit de meilleurs résultats que l'algorithme ACE.

2.5.1.3. La méthode de Foisel, Chevrier, Haton

Rémy Foisel [Foisel, 1997] propose un modèle dans lequel la structure d'organisation est un ensemble d'interactions privilégiées entre agents. Une interaction est décrite par sa motivation, les intervenants et par des protocoles. Un agent estime le bénéfice qu'il a à participer à une organisation en évaluant le gain qu'il retire de chacune de ses interactions. Lorsque ce gain est considéré comme non satisfaisant ou de peu d'intérêt, l'interaction impliquée est remise en cause. Une interaction peut être remise en cause pour deux raisons distinctes : soit le protocole suivi est jugé inadapté pour mener l'interaction, soit les participants ne satisfont pas correctement ou suffisamment la motivation qui la conditionne. Un agent doit donc être doté de capacités à évaluer les interactions qu'il accomplit et de moyens pour modifier le protocole d'interactions et/ou les accointances [Trouilhet, 1993] en fonction de l'évaluation réalisée. L'auteur se limite aux changements d'accointances (changement d'interaction).

L'évaluation d'une interaction correspond aux différents points de vue que l'agent peut avoir sur l'interaction : qualité de la réponse fournie, coût de cette réponse. Chaque agent mesure ses propres critères vis-à-vis d'une interaction et les évalue selon son point de vue à la fin de celle-ci.

L'évaluation de la qualité d'une interaction est fonction d'un vecteur de critères mesurables appelé c et fixé *a priori* pour chaque type d'interaction en fonction des protocoles. La qualité de l'interaction ($Q(c)$) est une fonction qui combine ces différents critères pour fournir un réel appartenant à l'intervalle $[0,1]$ et mesurant une distance qui se réfère à un idéal par rapport auquel la distance sera jugée proche ou non.

La qualification d'une interaction (ql) sert à choisir une des interactions alternatives ou une des sollicitations dont l'agent fait l'objet avant d'entamer les échanges. Deux catégories de qualifications sont définies : la confiance (exprime le bénéfice attendu de l'interaction et permet d'ordonner les interactions alternatives) et l'attention. Une fois l'interaction terminée, l'initiateur et l'agent sollicité évaluent chacun cette interaction (selon leurs points de vue et leurs critères).

Une interaction peut être remise en cause parce que les critères mesurés pendant celle-ci ne sont plus satisfaisants. Les participants sont alors jugés inadéquats par rapport à la qualité attendue de cette interaction. L'adéquation des participants est jugée de la manière suivante :

Si $Q(c) > t$ alors $ql = R(Q(c), ql)$ sinon $ql = W(Q(c), ql)$ avec t un seuil fixé, W et R sont respectivement deux fonctions d'affaiblissement et de renforcement de ql proportionnellement à la qualité $Q(c)$ de l'interaction.

Le mécanisme d'évolution des interactions est toujours actif pour réagir à toute perturbation.

Cet algorithme a été appliqué à la compétition dans un mécanisme de marché dans lequel un ensemble de 9 agents (2 producteurs, 1 consommateur et 6 producteurs/consommateurs) doivent gérer au mieux leur stock de marchandise (demande ou écoulement de stock). Les résultats obtenus

montrent que le modèle proposé améliore les performances globales et l'activité de production de la société, dote le système de capacité d'adaptation lors d'éventuelles perturbations et conserve une certaine stabilité de l'organisation.

2.5.1.4. Le modèle de Prasad, Lander, Lesser

Nagendra Prasad [Prasad, 1997] propose une méthode d'apprentissage sur l'organisation visant à prouver l'utilité de l'auto-organisation dans un système composé d'agents ayant recours à des interactions complexes. Selon elle, la connaissance organisationnelle peut être décrite par une spécification de la manière dont la recherche globale doit être organisée (en termes de rôles que les agents doivent jouer dans un processus de recherche : quelle information ils doivent communiquer, quand et à qui). Elle fournit aux agents un moyen de manipuler de manière efficace et sûre des tâches coopératives. Un rôle est décrit en tant que tâche ou ensemble de tâches à réaliser, (en utilisant éventuellement plusieurs opérateurs), dans le but d'atteindre un objectif. Les rôles sont attribués par rapport à un modèle. Un agent n'a pas à tenir tous les rôles ; le but de l'apprentissage consiste pour l'agent à apprendre le rôle qu'il doit tenir.

La méthode d'apprentissage proposée est appliquée au système L_TEAM dans lequel chaque agent est dédié à la résolution d'un sous problème et à l'obtention d'une solution partielle. Avant d'initier le processus de résolution pour atteindre l'objectif global cherché, chaque agent doit apprendre son rôle organisationnel. L'apprentissage consiste à négocier la distribution des tâches de sorte à ce que tous les agents soient satisfaits du rôle qu'ils jouent. Cette stratégie d'apprentissage repose sur une variante du formalisme UPC : à chaque tuple <état, rôle, état final> est associé un vecteur UPC (utilité, probabilité et coût) qui reflète l'activité effectuée par le système et les valeurs utiles pour le futur. Un agent applique la fonction d'évaluation à tous les rôles applicables dans l'état présent de la recherche et sélectionne le rôle qui maximise les estimations. Ces informations (utilité, probabilité, coût) permettent d'attribuer à chaque agent un rôle organisationnel qui soit en accord avec les caractéristiques de l'objectif global.

Le formalisme UPC classique a été modifié pour permettre un apprentissage organisationnel des rôles pour les agents. Tous les états possibles de la recherche sont rangés dans une classe finie de situations connues. Ces classes de situations représentent des abstractions de l'état de recherche. Ainsi, pour chaque agent, il y a un vecteur UPC par situation et par rôle menant à un état final. Une situation dans L_TEAM est représentée par un vecteur de caractéristiques dont les valeurs déterminent la classe d'un état de la recherche. Dans L_TEAM, un agent devant prendre une décision retrouve les valeurs UPC en fonction du vecteur de situation pour tous les rôles qui sont applicables dans l'état courant.

Une méthode d'apprentissage supervisé est utilisée pour apprendre à estimer les vecteurs UPC pour chacun des états. Les agents explorent collectivement l'espace des attributions de rôles possibles pour identifier de bonnes attributions de rôle dans chacun des états. L'attribution de rôle à un agent particulier est fonction de l'état de la résolution de problème des autres agents et également de la nature des espaces de recherche non locaux.

Connaître la pertinence d'un rôle est un processus plus compliqué nécessitant une certaine compréhension du système. Par exemple, dans L_TEAM, les agents peuvent appliquer des rôles qui mènent à des solutions infaisables en raison d'éventuels conflits de demande. Les rôles menant à un conflit qui sont suivis d'un échange d'information visant à le supprimer sont récompensés. L'algorithme d'apprentissage pour le potentiel d'un rôle utilise encore une approche supervisée et est similaire à celui de l'estimation de l'utilité.

Les expériences effectuées ont montré que les rôles organisationnels joués par les agents sont importants pour l'efficacité du processus de recherche et pour la qualité des solutions finales produites.

2.5.1.5. Le modèle de Terabe, Washio, Katai, Sawaragi

Masahiro Terabe [Terabe, 1997] propose un modèle qu'il applique au problème d'allocation de tâches. Le but organisationnel de l'allocation de tâches est d'obtenir une organisation permettant de minimiser les coûts d'exécution des tâches et les coûts de communication.

Le processus d'allocation de tâche s'effectue en cinq étapes. Tout d'abord des tâches, provenant de l'extérieur de l'organisation, arrivent à chaque agent avec une probabilité équivalente. L'agent qui reçoit la nouvelle tâche devient un manager de la tâche et choisit un agent (travailleur) parmi tous les agents (lui-même y compris) dans l'organisation. L'agent travailleur exécute la tâche et lorsqu'il a terminé, il l'indique au manager.

Il propose trois méthodes différentes pour allouer la résolution de la tâche à un agent. La première méthode est basée sur le réseau contractuel (CNP) de Smith [Smith, 1980]. Dans la seconde, le manager choisit l'agent travailleur en fonction de la connaissance qu'il possède sur les autres agents. Enfin, dans la troisième méthode le manager exécute la tâche seul.

Chaque agent possède un mécanisme d'apprentissage par renforcement. Le but de ce mécanisme est de permettre l'exécution de tâches en peu de temps. La connaissance du temps nécessaire aux autres membres pour effectuer un type de tâche donné est représentée par $e_{ij}(agent_j, task_k)$. Cette valeur est évaluée par le manager $agent_i$. Elle estime le temps dont un travailleur $agent_j$ a besoin pour effectuer un type de tâche $task_k$. Les agents ont aussi des croyances sur la précision de leur connaissance : $rel_i(task_k)$ montre l'évaluation de la précision de l'agent i sur la tâche $task_k$.

Le mécanisme d'allocation de tâches, choisi à un instant donné, est fonction de la connaissance du manager. Si sa précision est inférieure à un seuil donné, le manager choisit soit d'exécuter la tâche

lui-même, soit utilise le CNP. Il tranche selon le coût de communication. La seconde méthode est utilisée dans le cas contraire.

Il y a apprentissage, dans le CNP, lorsque le travailleur choisi a terminé l'exécution de la tâche et a communiqué à son manager la fin de son travail. L'apprentissage s'exprime par la formule : $et_j(\text{agent}_j, \text{task}_j) = (1 - \alpha) et_j(\text{agent}_j, \text{task}_k) + T\alpha$ où T est le temps consommé par le travailleur pour exécuter la tâche donnée, α est une évaluation de l'apprentissage. Si elle est voisine de 1 le manager change sa connaissance en fonction essentiellement des nouvelles expériences. Si le temps T est proche du temps estimé par le manager, la valeur de la précision de la connaissance est augmentée. Ce mécanisme d'apprentissage se retrouve également dans la seconde méthode. Dans le dernier cas, le manager se débrouille seul, il n'y a donc pas d'apprentissage sur l'organisation. Ces trois méthodes ont été comparées dans un système comportant 5 agents et 3 tâches à exécuter. Le CNP donne de meilleur résultat (en temps) que si le manager se débrouille seul. L'apprentissage sur l'organisation diminue les coûts totaux et les coûts de communication alors que les coûts d'exécution de tâche augmentent (en temps). Enfin le style d'allocation de tâche change suivant le processus d'apprentissage. Au début de l'apprentissage, les agents allouent la plupart des tâches en utilisant le CNP puis après un certain temps ils utilisent la seconde méthode et enfin se débrouillent par eux-mêmes.

2.5.2. La distribution des algorithmes

Les deux algorithmes d'apprentissage proposés par Weiß, issus de l'algorithme du "bucket brigade", sont distribués au sein des agents.

Le mécanisme d'apprentissage proposé par Shaw est composé de deux processus :

- un processus d'offre comme celui utilisé dans le réseau contractuel de Smith. Le processus de récompense est vu comme un mécanisme de coordination efficace pour la résolution de problème multi-agent.
- un processus de transformation génétique.

Le premier processus, mis en oeuvre à l'aide de systèmes à classifieurs, est distribué au sein de chacun des agents. En revanche, l'algorithme génétique ne l'est pas. Il est utilisé pour rendre le système adaptatif. Il combine des agents ayant une force élevée et remplace les plus faibles par cette combinaison. Il est donc nécessairement centralisé car il doit connaître la force des connaissances assignées à tous les agents du système.

Les méthodes proposés par Rémy Foisel, Prasad et Terabe sont distribuées au sein de chaque agent.

2.5.3. L'asynchronisme des algorithmes d'apprentissage proposés

L'algorithme génétique de Shaw nécessite la connaissance des forces de tous les agents pour savoir quelles règles combiner mais la création d'un nouvel agent se fait indépendamment du travail des autres agents. Il n'y a donc pas synchronisation de tous les agents lors de l'apprentissage proprement dit.

Tous les agents de Weiß ont besoin de connaître les forces de tous les agents pour apprendre dans un contexte donné. Il y a donc synchronisation du mécanisme d'apprentissage.

L'interlocuteur et le locuteur d'une interaction chez Foisel ne peuvent mettre à jour leur confiance en l'interaction qu'ils ont eue, qu'une fois celle-ci terminée. Cependant ils le font de manière totalement asynchrone l'un de l'autre.

Les agents de Prasad n'ont pas besoin de se synchroniser pour définir le rôle qu'ils doivent jouer à un instant donné.

Dans Terabe, les différents managers de tâche n'ont pas besoin de se synchroniser entre eux pour apprendre, ils ont en revanche besoin de se synchroniser individuellement avec l'agent à qui ils ont délégué la résolution de la tâche.

2.5.4. La coopération dans les algorithmes

Dans la méthode de Shaw, ce sont les agents, qui interviennent le plus dans le processus de résolution, qui sont choisis pour voir leurs capacités dupliquées car ils possèdent les forces les plus grandes. La coopération est le critère utilisé pour guider l'apprentissage.

L'apprentissage est parfois vu comme mécanisme étudié pour permettre aux agents de mieux se coordonner. Dans Weiß, les agents qui font la meilleure offre sont autorisés à effectuer leur action. Pour cela ils communiquent aux autres agents les actions qu'ils peuvent effectuer avec une offre associée dans un contexte donné. Ils essaient donc se coordonner au mieux.

On ne peut pas parler de coopération dans la méthode proposée par Rémy Foisel ; ceci dit, les liens d'interactions qui perdurent dans l'organisation sont le reflet d'une coopération entre agents.

La définition de rôle chez Prasad permet une résolution coopérative.

Dans Terabe, les agents coopèrent entre eux selon le modèle du Contract Net et d'une méthode simplifiée.

2.6. DISCUSSION

Dans ces quelques modèles, les auteurs montrent que l'apprentissage collectif permet au système d'être plus performant (du point de vue de son fonctionnement) qu'un système dans lequel l'apprentissage est réalisé de manière classique.

L'étude des diverses méthodes fait apparaître deux niveaux d'apprentissage. Les connaissances des agents s'enrichissent soit au niveau du domaine des compétences, soit au niveau des interactions. Dans le premier cas, le processus d'apprentissage utilise les interactions entre les agents pour permettre l'enrichissement des compétences. Dans le second cas, l'apprentissage évolue au fur et à mesure de l'activité du système et porte sur les liens d'interactions.

L'apprentissage classique en IA permet certes d'enrichir les connaissances relatives au domaine, mais le fait que l'agent interagisse avec d'autres lui permet d'apprendre plus sur le domaine qu'isolément. C'est le cas dans la méthode utilisée par Sian et par Sekaran. Les performances associées au fonctionnement du système, sont aussi améliorées par le fait que les agents se coordonnent ou interagissent mieux, comme avec la méthode de Weiß.

Dans la plupart des systèmes étudiés, les agents sont autonomes pour la résolution, mais dans les méthodes d'apprentissage présentées par Weiß et Shaw une synchronisation est requise pour apprendre. Ceci peut nuire à l'autonomie de l'agent.

Enfin, ces réalisations montrent aussi que tous les agents ont tendance à utiliser les mêmes principes de base pour les algorithmes d'apprentissage fondés sur le feed-back et le renforcement. Cependant, on peut noter que toutes les méthodes proposées font un présupposé sur la fonction qu'est sensée accomplir le système ce qui est contraire aux hypothèses que nous avons posées.

D'après les exemples d'apprentissage que nous venons d'étudier, il est possible de dégager les principales contraintes auxquelles doit répondre un mécanisme d'apprentissage qui soit relativement général.

Tout d'abord, il est essentiel que le processus d'apprentissage du système multi-agent soit implémenté de manière **distribuée** c'est-à-dire qu'il soit localisé chez les agents et qu'il opère à partir des informations connues localement par l'agent qui l'abrite. De plus, chaque agent doit pouvoir utiliser un processus d'apprentissage adéquat c'est-à-dire lié à la manière dont est implémenté l'agent. Le mécanisme d'apprentissage peut différer d'un agent à l'autre, ceci est d'autant plus réalisable qu'il est distribué. Comme nous l'avons vu dans les travaux exposés précédemment, les agents peuvent apprendre par observation des autres mais aussi en communiquant. Cette contrainte de distribution est en accord avec le fait que la distribution à la fois des données et du contrôle est de plus en plus présente dans les systèmes multi-agents actuels. Il peut sembler, au premier abord, que les méthodes d'apprentissage mises en oeuvre dans un tel contexte soient moins performantes que celles mises en oeuvre dans le cadre de sociétés où tous les agents ont une vue globale du système. Cependant elles sont excellentes du fait de l'incomplétude des connaissances.

D'autre part, l'activité d'apprentissage ne doit pas nuire à l'autonomie des agents. Cette autonomie leur permet d'avoir un fonctionnement asynchrone dans leurs activités de raisonnement, d'action sur l'environnement, de communication. Pour conserver cette notion d'asynchronisme au niveau de l'apprentissage, il faut que l'agent décide, en fonction de son état et de ses connaissances locales, de l'instant où il doit apprendre. Le processus d'apprentissage doit donc être **asynchrone**.

Enfin nous pensons que l'attitude **coopérative** est un principe général pour améliorer le résultat global de la société et définir un critère d'apprentissage efficace. A ce propos, Enric Placa & al. [Placa, 1996] ont montré les relations qui existent entre l'apprentissage et la coopération. Le parallélisme entre l'apprentissage et la coopération vient du fait que ce sont tous deux des moyens de pallier les déficits et les lacunes d'un agent. Ils résument ce parallélisme de la manière suivante : "Pourquoi avoir besoin d'apprendre ? Pour améliorer la performance individuelle, la précision (la qualité des solutions), l'efficacité (la vitesse pour trouver des solutions) et enfin le champ des problèmes solubles. Pourquoi avoir besoin de coopérer ? Pour améliorer la performance individuelle, la qualité des solutions, l'efficacité pour atteindre des solutions, et pour terminer des tâches que les agents ne peuvent résoudre seuls".

L'apprentissage ne peut donc selon eux être dissocié de la coopération et ils désirent étudier son apport sur le processus d'apprentissage. Pour notre part, nous allons étudier dans le prochain chapitre l'apport de la coopération sur la résolution de problèmes au travers de divers exemples.

Enfin, outre construire une méthode d'apprentissage ayant ces trois caractéristiques, nous souhaitons également que notre méthode soit indépendante de la finalité devant être rendue par le système. Elle doit seulement tenir compte de la nature des interactions entre les agents constituant le système.

CHAPITRE 3

L'AUTO-ORGANISATION : ÉTAT DE L'ART ET PRINCIPE

“St Thomas d’Aquin était probablement un des penseurs catholiques les plus influents. Il construisit des preuves logiques de l’existence de Dieu. Une de ces preuves se rapportait à Dieu comme étant l’organisateur ou le créateur suprême. L’argument était que tout devait être organisé et cela nécessitait un organisateur. A son tour, l’organisateur devait être organisé et ainsi de suite jusqu’à remonter à l’organisateur originel qui devait exister depuis la nuit des temps: il s’agissait de Dieu ! Si quelque chose est organisé, on tend à penser qu’une influence extérieure l’a organisé à un quelconque instant. Mais, cela n’est pas nécessaire! Le concept d’auto-organisation est important puisqu’il semble maintenant que la vie elle-même se produise selon un procédé d’auto-organisation, où des substances chimiques se regroupent de manière plus ou moins aléatoire, et s’organisent graduellement en entités vivantes. Bien sûr, il pourra toujours être argumenté que la présence de Dieu était nécessaire pour créer les substances chimiques!”
[Heylighen, 1995].

3.1. INTRODUCTION

Les recherches menées actuellement dans le domaine multi-agent s’intéressent particulièrement aux systèmes dotés d’une très forte dynamique et constitués de nombreux agents. Chaque agent n’a qu’une vue partielle du système dans lequel il est situé et évolue de manière autonome. Par autonomie nous sous-entendons, autonomie dans son comportement, dans sa prise de décision. Cela intègre le point de vue de Conte et Castelfranchi, qui définit l’autonomie comme étant “la capacité d’un agent à générer et poursuivre ses propres buts” [Conte, 1995].

Il est donc impossible, dans ce contexte, d’avoir une vision globale du système. La dynamique inhérente à ces systèmes provient des nombreuses créations, suppressions et modifications d’agents toutes décidées de manière autonome, ainsi que de la pression exercée par l’environnement dans lequel le système est immergé. Dès lors, on comprend aisément qu’il est utopique voire impossible de vouloir prévoir, dès la conception, toutes les situations pouvant survenir durant le fonctionnement du système car l’imprévu fait partie intégrante de la problématique. Parmi ces situations, certaines sont qualifiées de perturbatrices car elles dégradent l’activité collective de la société. Elles se produisent fréquemment parce que les agents n’ont qu’une connaissance incomplète voire erronée du monde et des autres.

Trois stratégies possibles peuvent être mises en pratique pour faire face aux perturbations imprévues. La première stratégie consiste à les éviter dès la conception du système, ce qui est quasiment irréalisable. La seconde stratégie consiste à les masquer, les détruire ou les inhiber. La dernière stratégie consiste à intégrer la notion de hasard dans le système et à considérer que la perturbation “fait partie” du système : le système DOIT s’y adapter. Cette approche conduit à l’élaboration de systèmes robustes, adaptatifs et peu sensibles au “bruit”. Quelquefois même, le bruit peut contribuer à “enrichir” le niveau d’organisation du système : “fluctuations auto-catalytiques, oscillations et cycles peuvent conduire à la naissance de structures organisées à partir du désordre. Ce processus est appelé par Von Foerster ‘order from noise’ (ordre à partir du bruit). Pour I. Prigogine, c’est ‘l’ordre par fluctuation’. Les structures organisées transformant l’énergie sont des ‘structures dissipatives’”[de Rosnay, 1975].

Nous nous plaçons dans ce troisième et dernier cas pour étudier le concept d’auto-organisation. Plus précisément, nous nous intéressons aux systèmes auto-organisateurs pour lesquels la finalité n’est pas *a priori* connue ; autrement dit, les agents ne s’auto-organisent pas pour garantir l’obtention d’un résultat particulier mais pour s’adapter uniquement aux perturbations extérieures. Les éléments constituant le système poursuivent un objectif individuel et non global. La collectivité, quant à elle, fournit un résultat global qui sera jugé par un observateur extérieur au système connaissant les raisons pour lesquelles le système a été conçu. Celui-ci est à même de statuer sur l’adéquation entre le résultat fourni et la fonction du système.

Avant d’étudier les caractéristiques de l’auto-organisation ainsi que ses diverses méthodes de mise en oeuvre dans le cadre de la vie artificielle et des systèmes naturels, il est nécessaire de bien définir le concept d’auto-organisation ou de système auto-organisateur.

Ce chapitre s’organise en neuf parties : après avoir donné les différentes définitions de l’auto-organisation qu’il est possible de trouver dans la littérature, nous présenterons quelques notions fondamentales, telles que les caractéristiques d’un système, l’organisation d’un système et les caractéristiques des systèmes auto-organisateurs. Nous étudierons ensuite l’auto-organisation dans les domaines des systèmes naturels, de la systémique, des modèles connexionnistes, des algorithmes génétiques, de la robotique et enfin de l’intelligence artificielle distribuée. Ce chapitre se termine par la mise en évidence des liens existant entre l’auto-organisation et la notion d’émergence.

3.2. QUELQUES DÉFINITIONS

L'auto-organisation a été une notion initialement étudiée et appliquée dans les secteurs de la biologie, de la physique et de la chimie. Elle est également étudiée dans les domaines de la systémique, de la vie artificielle et des systèmes naturels, mais n'est qu'à l'état de recherche dans le domaine de l'informatique et particulièrement dans celui de l'Intelligence Artificielle. Luc Steels encourage grandement son étude: "L'auto-organisation est maintenant comprise pour être un processus fondamental en biologie (particulièrement dans l'évolution, la formation de modèle, ou le système immunitaire), en physique et en chimie. Il n'y a pas de raison pour que cela ne puisse pas être aussi pertinent en IA" [Steels, 1990].

“L'objectif de l'auto-organisation est de permettre l'évolution dynamique de l'existant en fonction du contexte de façon à assurer la viabilité du système” [Camps, 1994]. L'auto-organisation est une aptitude nécessaire dans les applications où l'on désire avoir une grande tolérance aux pannes (arrêt d'un ordinateur...) ou lorsque le système est très complexe. Elle permet aux entités composant le système de s'adapter à leur environnement soit par spécialisation des fonctions (apprentissage), soit par modification de la topologie du groupe ce qui donne essence à un nouveau modèle d'organisation.

La plus ancienne définition concernant le concept d'auto-organisation, proposée en 1954 par Farley et Clark du laboratoire Lincoln, a été citée et reprise dans le travail de Cem Ünsal [Ünsal, 1993] :

✍ “Un système auto-organisateur est un système qui change sa structure de base comme fonction de son expérience et de son environnement.”

La définition proposée par Francisco Varela est la suivante [Varela, 1988] :

✍ “L'auto-organisation est une description d'un comportement, elle a une valeur heuristique et elle permet d'indiquer un phénomène. Elle est condamnée à rester une simple description, tant qu'on ne se préoccupe pas de rechercher le mécanisme qui est à son origine.”

La définition proposée par Krippendorff [Krippendorff, 1997] est plus concrète :

✍ “L'auto-organisation est un processus où l'organisation (contrainte, redondante) d'un système croît de manière spontanée, i.e. sans que cet accroissement soit contrôlé par l'environnement ou ce qui l'entoure ou encore un système externe.”

Pierre Marcenac [Marcenac, 1997a], définit ainsi l'auto-organisation :

✍ “L'auto-organisation définit la propriété d'un système qui s'organise ou se réorganise dans le temps, pour former des structures sémantiquement remarquables”.

Éric Bonabeau [Bonabeau, 1997] propose une définition très explicite qui introduit le concept d'émergence :

↳ “L'auto-organisation caractérise tout processus au cours duquel des structures émergent au niveau collectif (ou plus généralement apparition d'une structure à l'échelle N+1 à partir d'une dynamique définie à l'échelle N), à partir de la multitude des interactions entre individus, sans être codées explicitement au niveau individuel”.

Le groupe de travail MARCIA [MARCIA, 1996] s'est intéressé au concept d'auto-organisation. La définition qu'il en a donné est la suivante : “l'auto-organisation correspond à une réorganisation décidée de manière autonome par les agents au sein du système. [...] L'auto-organisation d'un système s'est produite si le système a changé de structure”.

La définition proposée par notre équipe [Piquemal, 1996b] est conforme à celle proposée par Varela [Varela, 1993]. L'auto-organisation d'un système consiste en la transformation de la topologie (c'est-à-dire des connections du réseau) de ses parties en tant que résultat du fonctionnement de ce même réseau dans le cadre du couplage structurel avec l'environnement. Dans un système auto-organisateur, les règles d'organisation sont intérieures au système, qui apparaît informationnellement clos. Les systèmes auto-organiseurs appartiennent à la classe des systèmes autonomes (systèmes spécifiés par des mécanismes internes d'auto-organisation) et non pas hétéronomes (définis par des mécanismes extérieurs de contrôle).

Toutes ces définitions introduisent plusieurs concepts, chacun étant fondamental pour la description d'un tel phénomène : l'organisation, l'apparition de structure, l'interaction, le non contrôle et l'émergence. Nous reviendrons sur ces notions tout au long de notre analyse.

Avant d'étudier la mise en oeuvre de l'auto-organisation dans des domaines aussi divers que la robotique, les sociétés animales... nous allons introduire quelques notions essentielles.

3.3. Notions fondamentales

Nous allons, dans ce paragraphe, expliciter quelques notions qui sont importantes pour comprendre certains aspects de l'auto-organisation. Ces notions concernent notamment les aspects liés au système, à l'organisation d'un système, aux interactions du système et de son environnement.

3.3.1. Caractéristiques essentielles d'un système

Un système est plongé dans un **environnement** duquel il ne peut être dissocié et avec qui il est amené à interagir. L'environnement représente tout ce qui est extérieur au système. Les rapports d'un système avec son environnement sont matérialisés par des entrées et des sorties. Ces rapports sont plus ou moins nombreux et intenses selon que le système est dit fermé ou ouvert. Les systèmes fermés vivent totalement repliés sur eux-mêmes. Les systèmes ouverts, quant à eux, effectuent de nombreux échanges avec tout ce qui les entoure, donc leur environnement. Un système et son environnement entretiennent des échanges réciproques qui leur permettent de s'ajuster mutuellement (cf. §4.2.3.). L'interaction est donc une notion qui revêt une grande importance.

Joël de Rosnay [de Rosnay, 1975] et Daniel Durand [Durand, 1979], tous deux systémiciens, proposent une double description pour un système : une description structurelle et une description fonctionnelle.

Au niveau structurel, ils dénombrent essentiellement quatre éléments entrant dans la constitution d'un système :

- La **frontière** qui le sépare de son environnement. Cette frontière est plus ou moins perméable. Dans le domaine informatique, elle désigne par exemple l'interface homme/machine d'un système. D'un point de vue biologique, il s'agit de la membrane d'une cellule.
- Les **éléments** qui peuvent être identifiés, dénombrés et classés. Ces éléments sont plus ou moins hétérogènes en fonction du système étudié. Dans le cadre des systèmes multi-agents, il s'agit des agents.
- Le **réseau** de relation, de transport et de communication qui véhicule soit des matières solides, liquides ou gazeuses, soit de l'énergie, soit des informations sous toutes les formes possibles. Dans les systèmes multi-agents, ce réseau de relations est représenté par les liens de communication inter-agents sur lesquels se fondent peu à peu les connaissances sur autrui des agents.
- Les **réservoirs** dans lesquels sont stockés des matières, de l'énergie, des produits, de l'information.

Au niveau fonctionnel, il distingue également quatre composants pour un système :

- des **flux** de natures diverses : de matières, de produits, d'énergie, de monnaie, d'informations... Ils circulent dans les divers réseaux et transitent dans les réservoirs du système. Dans le domaine informatique, il s'agit des signaux échangés entre entités.

- des **centres de décision** qui reçoivent les informations et les transforment en actions, en agissant sur les débits des différents flux. En considérant toujours que l'on se place dans un univers multi-agent, ces centres de décision sont une partie des compétences et des aptitudes des agents. Ces derniers jouissent d'une autonomie suffisante pour prendre par eux-mêmes les décisions opportunes.
- des **boucles de rétroaction** qui ont pour objet d'informer les déclencheurs de ce qui se passe en aval donc de leur permettre de prendre leurs décisions en connaissance de cause. Il s'agit là d'un comportement dit non linéaire. Le mécanisme de rétroaction est en particulier utilisé pour mettre en oeuvre les apprentissages par renforcement.
- enfin, des **délais de réponse** qui permettent de procéder aux ajustements de temps nécessaires à la bonne marche du système.

Une forme particulière d'interaction est la **rétroaction** (ou feed-back) dont l'étude est au centre des travaux de la cybernétique. "L'approche cybernétique distingue de manière stricte les propriétés de rétroaction qui caractérisent certaines étapes des transformations dont un système est le siège, et les propriétés fonctionnelles globales de ce système, comme la stabilité ou la régulation [Stengers, 1983]. Lorsqu'un système interagit avec son environnement, les entrées sont le résultat de l'influence de l'environnement sur le système et les sorties sont l'influence du système sur l'environnement. Lors d'une rétroaction, l'information sur le résultat de la transformation induite par les entrées est transmise au système sous la forme d'une donnée d'entrée.

Lorsque les nouvelles données agissent dans le même sens que l'action principale, c'est-à-dire amplifient la transformation, la rétroaction est dite positive ; ses effets sont cumulatifs. En revanche, si les nouvelles données produisent un résultat dans le sens contraire à celui fourni par les précédents résultats, la rétroaction est dite négative ; ses effets stabilisent le système. Comme l'illustrent les figures ci-dessous de Joël de Rosnay [de Rosnay, 1997], une rétroaction positive entraîne une croissance ou une déclinaison exponentielle alors qu'une rétroaction négative entraîne un maintien d'équilibre.

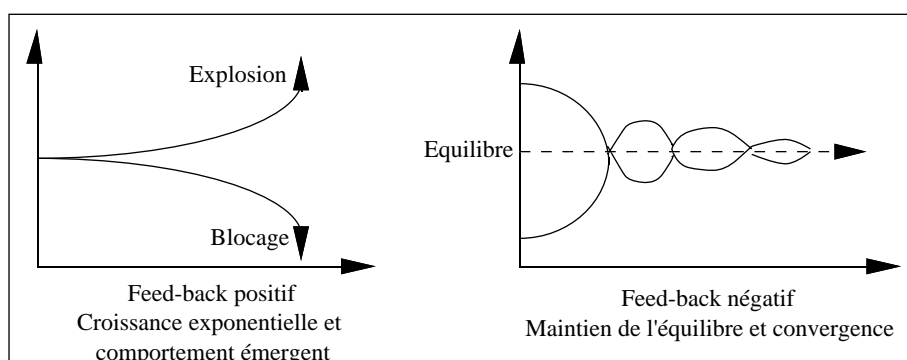


Figure 3.3.1. : Les deux types de rétroaction possibles

Un système est dit **complexe**, “s’il est constitué d’une grande variété de composants ou d’éléments possédant des fonctions spécialisées. Les éléments sont organisés en niveaux hiérarchiques internes (par exemple, dans le corps humain: les cellules, les organes, les systèmes d’organes). Les différents niveaux et éléments individuels sont reliés par une grande variété de liaisons. Il en résulte une haute densité d’interconnexions. Les interactions entre les éléments d’un système complexe sont d’un type particulier. On dit que ces interactions sont non linéaires” [de Rosnay, 1975].

La complexité d’un système peut être caractérisée par un désordre apparent dans lequel existe un ordre que nous ne connaissons pas. Le comportement de tels systèmes est imprédictible, il ne fait pas appel à des comportements linéaires. Une des caractéristiques des systèmes complexes est de s’adapter à des contraintes extérieures en adoptant une nouvelle phase d’équilibre (attracteur). En situation de non équilibre, le système est alors capable de transiter d’un état d’équilibre dans lequel il se trouvait vers un autre (attracteurs fractals). Deux catégories d’attracteurs peuvent être distinguées [Perez, 1988] :

- l’attracteur simple : le système est “attiré” vers un régime stable (pendule ou liquide chaud qui se refroidit). Ces systèmes sont totalement déterminés par leurs conditions initiales et “insensibles” aux perturbations dans le sens où ils les “oublent”. Ces attracteurs n’ont donc qu’un petit nombre de points fixes.

- l’attracteur fractal : ici, au contraire, le système va “errer” parmi ses milliards de points fixes différents ; ces systèmes sont vulnérables aux conditions initiales mais également très sensibles aux perturbations.

Un système est avant tout, comme nous l’avons vu, composé d’éléments. Cela n’implique pas, comme on pourrait le croire au premier abord, qu’il est la somme de ses éléments. Comme l’écrivait déjà Pascal dans les Pensées, “je tiens pour impossible de connaître les parties sans connaître le tout, non plus que de connaître le tout sans connaître particulièrement les parties.” En effet, considérons un problème complexe dont la résolution nécessite plusieurs experts de différents domaines. Certes, chacun des experts possède un certain savoir-faire dans le domaine qu’il maîtrise mais l’agrégation des connaissances de chacun permet de résoudre un problème que chaque entité n’aurait pu résoudre isolément. Comme l’a montré Von Bertalanffy [Bertalanffy, 1993], “un système est un tout non réductible à ses parties”. Le tout est davantage qu’une forme **globale**, il implique l’apparition de qualités émergentes que ne possédaient pas les parties (cf. notion d’émergence au §3.10).

3.3.2. Organisation et auto-organisation d'un système

Le type de relation entre entités composant un système, autrement dit l'organisation d'un système, est importante lors de l'élaboration d'une solution à un problème donné. Mais qu'est ce qu'une organisation ?

L'organisation est considérée comme un concept fondamental de la systémique. Selon Maturana et Varela [Maturana, 1992], "**l'organisation** désigne les relations qui doivent exister entre les composants d'un système pour qu'il soit membre d'une classe donnée." Ils différencient cette notion du concept de **structure** qui, pour eux, "désigne les composants et les relations qui constituent une unité particulière et concrétise son organisation". La modification de la structure d'un système peut influencer l'organisation du système mais pas dans tous les cas. Notre définition de l'organisation d'un système est conforme à celle de Maturana et Varela puisqu'elle désigne l'ensemble des liens d'interaction possibles entre entités le constituant. Elle restreint l'autonomie des agents dans le sens où elle leur indique avec qui ils peuvent interagir.

Tout comme un système, une organisation possède un aspect statique : sa structure organisationnelle, et un aspect dynamique : le processus qui construit cette structure [Ferber, 1995], [MARCIA, 1996].

Une structure correspond à un état stable du système, c'est-à-dire à une organisation particulière. L'auto-organisation permet donc le passage, de manière autonome, d'une structure stable à une autre structure stable. La stabilité de la structure d'un système est fonction de la durée pendant laquelle cette dernière demeure stable malgré les perturbations tendant à la déstabiliser. L'auto-organisation met parfois en évidence des phénomènes de convergence vers des structures particulières. Ce phénomène se rapproche alors de la notion d'attracteur.

Il est essentiel de souligner qu'une organisation dépend fortement du problème en cours de résolution : elle est **contextuelle**. Elle peut être adéquate pour résoudre un problème donné mais s'avérer inadéquate pour un autre problème. C'est d'ailleurs ce qu'exprime W. Ross Ashby [Ashby, 1962] lorsqu'il dit : "there is no such thing as 'good organization' in any absolute sense. Always it is relative and an organization that is good in one context or under one criterion may be bad under another".

Il est intéressant d'envisager que les systèmes informatiques puissent se réorganiser de manière dynamique vis-à-vis de problèmes qui, eux aussi, peuvent évoluer au cours du temps, sans l'intervention d'opérateur extérieur : autrement dit, construire des systèmes auto-organiseurs. Un tel système pourrait alors s'adapter au contexte courant donc aux éventuelles perturbations par apprentissage. Nous considérons qu'un système **s'adapte**, si face à une situation imprévue par le concepteur de l'application finale, il ne se bloque pas mais réagit. Le système peut alors modifier son comportement face aux évolutions de son environnement sans en tirer de conséquences

durables. Ainsi, pour deux situations identiques rencontrées au cours de son histoire, le système adaptatif se comporte exactement de manière identique. Outre s'adapter, un système **auto-organisateur** apprend. En univers multi-agent, un système est dit auto-organisateur s'il modifie sa topologie au cours du temps mais aussi si chaque agent peut modifier ses compétences, ses croyances et ses attitudes sociales en fonction de l'instant courant et de l'expérience passée. L'**apprentissage** est omniprésent car il y a apparition de notions telles que la mémoire, l'expérience. Le système apprend à s'organiser en fonction de l'expérience passée pour permettre d'arriver plus vite à l'optimum qui est la meilleure organisation répondant au problème posé.

Plus, concrètement [Glize, 1997], [Piquemal, 1996b], "un système auto-organisateur est un système qui possède des états d'équilibre (attracteurs) correspondant à des organisations plus ou moins stables. Chaque organisation possède un bassin d'attraction correspondant à un ensemble de conditions initiales différentes. Lorsque ces conditions sont vérifiées, elles font converger le système vers l'organisation correspondante". La plupart du temps, le système auto-organisateur se trouve en l'un ou l'autre de ses états d'équilibre. Il passe d'un état d'organisation à un autre sous la pression désorganisatrice de son environnement. Plus ses états d'équilibre sont attractifs, moins le système passe de temps à transiter d'un état à un autre pour se stabiliser. Les phénomènes d'auto-organisation ainsi que les transitions vers des états attracteurs sont induits par les boucles de rétroaction. Ces dernières permettent alors au système auto-organisateur d'apprendre.

3.3.3. Caractéristiques des systèmes auto-organisateurs

Les systèmes auto-organisateurs sont particulièrement bien adaptés aux systèmes de nature distribués, fortement dynamiques, où aucun élément entrant dans sa constitution, ni même le concepteur, ne possède une vue globale du système. Contrairement aux systèmes déterministes, la panne ou la suppression d'un élément dans un système auto-organisateur affecte peu (voire pas du tout) le comportement de celui-ci. Il s'adapte de lui-même, de manière autonome, aux perturbations en provenance de son environnement.

D'autre part, les systèmes auto-organisateurs ont la faculté de réagir de manière non prédonnée à plusieurs perturbations similaires. Ils ont par conséquent la faculté d'apprendre au fur et à mesure qu'ils évoluent. L'être humain est l'exemple typique d'un système auto-organisateur... Comme le souligne et le schématise Daniel Durand [Durand, 1979], "l'homme construit sa propre liberté malgré une double contrainte : celle de l'environnement dans lequel il est immergé et celle du code génétique dont il hérite et qui détermine, au moins partiellement son individualité. Il construit cette liberté grâce à l'autonomie dont il jouit et à l'auto-organisation qu'il pratique en permanence. Cette aptitude à auto-organiser son cerveau pendant la période de l'enfance offre ensuite à l'homme

d'immenses possibilités et lui ouvre les voies royales de la culture, de la conscience et de la liberté, lui permettant ainsi d'accéder au plus haut niveau des systèmes : l'auto-finalisation.”

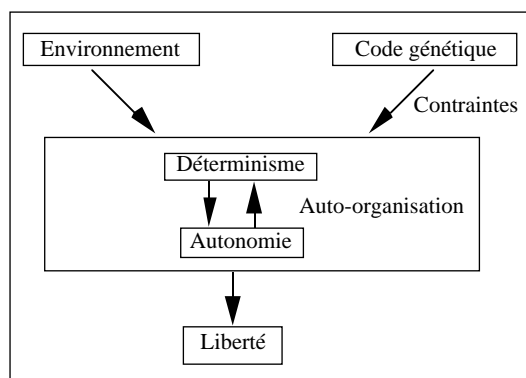


Figure 3.3.3 : L'auto-organisation chez l'être humain

Une autre caractéristique essentielle et particulièrement intéressante des systèmes auto-organisateur est qu'ils peuvent accomplir des tâches complexes en utilisant des comportements individuels simples et concis. L'auto-organisation ne requiert pas de mécanismes compliqués pour sa mise en oeuvre puisque c'est un phénomène spontané et imprévisible mais elle fait apparaître des structures qui, elles, peuvent paraître complexes.

Les systèmes auto-organisateur sont capables d'effectuer des tâches complexes qu'aucun élément entrant dans sa constitution n'aurait pu effectuer seul. La notion d'émergence est donc très fortement liée à la propriété d'auto-organisation (cf. §3.10).

En dépit de ses avantages évidents, l'auto-organisation n'est usitée de nos jours que dans très peu de domaines. Étant donnée l'hétérogénéité de ces domaines (informatique, physique, biologie...), la présentation de l'auto-organisation ne sera pas exhaustive. Nous présenterons seulement les principaux mécanismes de mise en oeuvre. Nous nous intéresserons pour cela aux domaines de la vie naturelle, de la vie artificielle, de "l'intelligence collective" (robotique, intelligence artificielle distribuée) et enfin de la systémique.

3.4. L'AUTO-ORGANISATION DANS LES SYSTÈMES NATURELS

L'auto-organisation chez les insectes sociaux est proche de celle que nous préconisons car les entités qui composent le système poursuivent un objectif individuel et ne cherchent pas à atteindre un but final déterminé. Elles s'adaptent aux éventuelles perturbations extérieures qui peuvent survenir durant leur travail et apprennent au fur et à mesure sur leur comportement. Le processus

d'auto-organisation qui est alors mis en oeuvre, ne nécessite pas de connaître la finalité que doit produire le système et est indépendant du domaine d'application.

Selon Guy Theraulaz, "l'éthologie est sans doute le domaine d'investigation où les processus d'auto-organisation peuvent d'un point de vue expérimental être le plus facilement étudiés. En effet, les règles de comportement de chaque individu au sein du groupe sont beaucoup plus facilement accessibles que dans n'importe quel système physico-chimique dans lesquels ces phénomènes similaires de structuration spontanée ont pourtant été initialement mis en évidence" [Theraulaz, 1997].

Les travaux sur l'auto-organisation des sociétés animales nécessitent de comprendre les comportements individuels et inter-individuels qui permettent l'apparition et le maintien d'une dynamique collective alors qu'un agent est incapable de penser cette dynamique. L'exemple le plus connu et dans lequel le phénomène d'auto-organisation apparaît le plus clairement et le plus simplement est celui du fourragement chez les fourmis.

Nous présenterons dans ce paragraphe les mécanismes d'auto-organisation mis en oeuvre chez les fourmis puis ceux utilisés par les termites.

3.4.1. Auto-organisation par auto-catalyse

Les expériences sur l'auto-organisation des fourmis en quête de nourriture ont mis en évidence un phénomène intéressant : un individu qui découvre une source de nourriture informe ses congénères de sa découverte en déposant sur le sol une substance chimique, un marquage phéromonal en retournant à son nid. Cette piste guide les autres individus de la société, qui eux-mêmes, après s'être alimentés renforcent le marquage de la piste lors du retour au nid et recrutent à leur tour. Ce phénomène tend à s'inverser au fur et à mesure de l'épuisement de la ressource car les fourmis récoltent et recrutent de moins en moins.

Malgré le marquage, certaines fourmis arrivent à se perdre. Le système de recrutement peut alors paraître inefficace, mais il n'en est rien car les fourmis perdues, en explorant aléatoirement l'espace vont alors peut-être trouver de nouvelles sources de nourriture et ainsi recruter à leur tour.

Des études ont été faites concernant les états d'équilibre vers lesquels tend un système qui repose sur un tel recrutement. Si on suppose un système comportant deux ressources de quantité égale, trois types de comportements peuvent être observés en faisant varier le nombre de fourmis :

- aucun recrutement n'est possible lorsque le nombre de récolteuses est trop faible,
- un recrutement apparaît à partir d'une valeur critique correspondant à un certain nombre de fourmis. Ce recrutement conduit à une exploitation symétrique des deux sources et se maintient jusqu'à ce qu'une nouvelle valeur critique du nombre d'individus apparaisse, auquel cas un nouveau comportement collectif est observé,

- lorsque le nombre de fourmis est encore plus important, une bifurcation se produit et la seule situation stable est alors une exploitation asymétrique des deux ressources, l'une étant fortement exploitée, l'autre faiblement. Cette distinction est d'autant plus flagrante que le nombre de fourmis croît.

Cet exemple montre que le marquage phéromonal introduit un certain ordre au sein de la colonie, appelé "ordre par fluctuation" par Nicolis et Prigogine [Prigogine, 1994]. Il permet de décrire l'évolution du système vers un nouveau régime dynamique où existent un ordre temporel et un ordre spatial entre les diverses fourmis : le système a un comportement "**holiste**". On peut également noter le caractère **adaptatif** du système, lorsqu'une nouvelle source de nourriture est découverte, lors de l'égaré d'une fourmi... L'aléatoire ou plus exactement le **chaos** est, dans ce cas là, à l'origine de l'auto-organisation de la colonie. En effet, les fourmis qui s'écartent du chemin marqué par les phéromones mettent le système hors de son état d'équilibre. Le sous-ensemble du système ayant quitté l'état d'équilibre peut, à son tour, constituer une structure organisée en particulier lorsque les fourmis trouvent une nouvelle source de nourriture. Une forme organisée peut donc naître de mouvements chaotiques aléatoires pour ensuite se stabiliser.

L'auto-organisation repose aussi implicitement sur la **coopération** entre fourmis, cette coopération étant mise en oeuvre par le biais de la communication indirecte. Elle est régie par un mécanisme **auto-catalytique** élaboré à partir d'un processus de **rétroaction**. La rétroaction positive permet d'amplifier le fourragement vers la ressource découverte ; elle devient négative lorsque la source s'amointrit.

Enfin notons que le **nombre d'entités** influe sur la capacité du système à s'auto-organiser.

3.4.2. Auto-organisation par stigmergie

Les termites ont également une aptitude à l'auto-organisation mis en oeuvre par un processus appelé stigmergie. L'édifice d'une loge royale chez les termites, fait apparaître un mécanisme curieux : après une phase durant laquelle les termites se déplacent de manière non coordonnée, certaines d'entre elles récoltent de la terre et façonnent des boulettes qu'elles déposent de manière aléatoire sur le sol. Lorsque les boulettes déposées atteignent une certaine densité, une coordination au niveau des dépôts de boulettes fait apparaître la construction de la loge royale. Les boulettes étant fraîchement imbibées de salive attirent d'autres termites qui déposent alors leur boulette sur le sommet du tas déjà existant de sorte à le transformer en pilier. Lorsque les piliers ainsi bâtis ont atteint une certaine hauteur, les termites déposent alors leur boulette latéralement, c'est-à-dire au bord du sommet de la section supérieure ; les boulettes ainsi déposées deviennent le point de départ d'une arche unissant deux piliers voisins.

Tout comme pour les fourmis, des études ont été faites concernant les états d'équilibre vers lesquels tend un système basé sur un tel comportement. Deux types de comportement ont été observés en fonction du nombre de termites :

- lorsque le nombre de termites participant à l'élaboration du nid est faible, la concentration en phéromone est tel qu'il ne peut influencer le comportement d'autres insectes avant de disparaître. La phase de dépôt aléatoire persiste alors aussi longtemps que le nombre de termites demeure peu élevé ;

- au fur et à mesure que le nombre d'insectes participant à la construction augmente, l'accroissement du taux de dépôts sur des sites préférentiels accroît la concentration en phéromone ce qui attire alors de plus en plus d'insectes, d'où le taux de dépôts... ainsi de suite jusqu'à atteindre un seuil où l'apparition de "patterns" traduit un mouvement coordonné des termites bâtisseuses. Le résultat est l'émergence de piliers due à une amplification rapide des dépôts de boulettes aux points de plus haute concentration en phéromone.

Le comportement des termites pour la construction des arches est régulé par le même phénomène.

Tout comme dans l'exemple précédent, le marquage phéromonal introduit un élément de coordination entre les termites bâtisseuses et l'organisation du déplacement des insectes lors de leur activité bâtisseuse. On peut observer un autre phénomène original : l'activité de construction des termitières ne dépend pas des ouvrières mais de l'état de la construction elle-même. L'insecte ne dirige pas son travail mais au contraire il est dirigé par lui : cette stimulation est appelée **stigmergie**. L'auto-organisation repose donc là encore sur la **coopération** entre termites. Cette coopération est mise en oeuvre par le biais de la communication indirecte. L'auto-organisation s'appuie aussi sur le mécanisme de **rétroaction** positive (auto-catalyse) permettant d'amplifier la construction de piliers puis d'arches. Le **chaos** intervient au tout début de la construction lors du dépôt des premières boulettes ; malgré ces fluctuations aléatoires, "la régularité de la structure produite apparaît spontanément dans le système sans qu'aucune "mesure de distance" entre deux piliers ne soit nécessaire aux insectes pour la produire" [Deneubourg, 1977]. Enfin tout comme précédemment le **nombre d'entités** doit être suffisant pour que le système puisse s'auto-organiser.

Il est important de souligner que ni les fourmis, ni les termites n'ont conscience d'édifier des organisations si élaborées : elles poursuivent individuellement un objectif propre. L'auto-organisation n'est donc observable que de l'extérieur du système. Tout se passe pour l'observateur comme s'il existait un agent coordinateur au centre de la colonie qui assurait les différentes activités des individus. La notion d'observateur est très importante dans les phénomènes d'auto-organisation car c'est lui seul qui peut juger de l'apparition de structure.

3.5. L'AUTO-ORGANISATION DANS LE DOMAINE DE LA SYSTÉMIQUE

Actuellement, la systémique tente de “comprendre les systèmes comme des entités auto-organisatrices, dont le fonctionnement et l'évolution sont les produits du comportement d'un ensemble d'entités en interaction” [Ferber, 1995]. Maturana et Varela [Maturana, 1992] ont alors introduit à cet effet les notions d'autopoïèse et d'homéostasie que nous allons détailler.

3.5.1. L'autopoïèse et l'auto-organisation

Un système est autopoïétique s'il est capable à la fois de “maintenir son organisation malgré les perturbations environnementales ainsi que les changements structurels et de régénérer ses composants au cours de ses opérations.” [Maturana, 1995]. En d'autres termes, un système est autopoïétique s'il est capable de s'auto-reproduire tout en maintenant son organisation en dépit des changements induits par ses interactions. Comme tout système vivant, l'homme est un système autopoïétique : il est capable de maintenir son organisation et de s'auto-reproduire.

Les interactions entre composants d'un système dépendent de l'organisation existante puisque celle-ci décrit les relations entre composants. Mais chaque composant d'un système agit sur l'environnement, sur autrui et également sur l'organisation en fonction des interactions qu'il a avec les autres composants du système. Les interactions sont donc à la fois le fruit et la base de la genèse de l'organisation d'un système. Une clôture opérationnelle est faite entre le système (qui conserve son identité) et l'organisation (qui change).

L'adaptation est une aptitude nécessaire à un système autopoïétique pour que les interactions avec son environnement ne soient pas destructrices. Si un système adaptatif est en outre capable d'auto-reproduction (dans laquelle l'auto-organisation est une aptitude indispensable) il sera qualifié d'autopoïétique.

3.5.2. L'homéostasie et l'auto-organisation

Selon Cannon [Durand, 1979], l'homéostasie décrit “l'ensemble des processus organiques qui agissent pour maintenir l'état stationnaire de l'organisme, dans sa morphologie et dans ses conditions intérieures, en dépit des perturbations extérieures.” Il s'agit du processus de régulation des systèmes vivants. Le maintien d'un système vivant s'effectue par un renouvellement constant de ses composants sans que son identité ne soit changée. Un exemple typique est le maintien de la température interne d'un être vivant.

L'homéostasie [Von Bertalanffy, 1993] peut être mise en oeuvre en utilisant des mécanismes de rétroaction négative. En effet, en générant des transformations allant à l'encontre des changements survenus, la rétroaction négative tend à stabiliser le système.

L'homéostasie est un processus inné dans les systèmes naturels car il est pour eux un moyen de survie. Contrairement aux machines artificielles qui perdurent malgré une interruption, l'arrêt de l'homéostasie est synonyme de mort du système. Dans les systèmes artificiels, l'homéostasie permet au concepteur d'éviter les situations extrêmes tout en lui indiquant comment maintenir l'équilibre.

Un système possédant un grand nombre d'attracteurs est très sensible au bruit. Il passe donc la plupart de son temps à s'auto-organiser en transitant d'une organisation à une autre en fonction des perturbations perçues. Il doit être capable au moment opportun, de converger vers un attracteur lui permettant de commuter vers une organisation pour revenir à un état stable. Le maintien d'un état stable (homéostasie) est une propriété essentielle des systèmes auto-organiseurs surtout s'ils possèdent un nombre potentiellement important d'états attracteurs.

3.6. L'AUTO-ORGANISATION DANS LES MODÈLES CONNEXIONNISTES

Les modèles connexionnistes [Bourret, 1991] sont fondés sur la métaphore du cerveau et sont caractérisés par trois composants :

- un réseau constitué d'éléments tous identiques appelés neurones formels,
- une règle d'activation locale à chaque neurone qui est fonction des valeurs d'entrée reçues d'autres neurones,
- une règle d'apprentissage qui permet de modifier les poids des connexions entre neurones.

Les neurones sont reliés entre eux par des connexions pondérées (poids synaptique). Si le poids d'une connexion est positif le lien est activateur, dans le cas contraire il est inhibiteur. Un neurone possède deux états : il est soit actif soit inactif. Chaque neurone met périodiquement à jour son état. Il effectue pour cela une sommation des signaux transmis par les neurones voisins. Si elle dépasse un certain seuil, le neurone devient actif et propage à son tour son potentiel d'action. Si la sommation est inférieure à ce seuil, le neurone reste inactif.

Il existe un grand nombre de modèles de réseau de neurones. Les plus classiques sont les modèles perceptrons multi-couches dans lesquels chaque neurone reçoit les signaux de la couche n-1 et transmet le résultat de ses traitements aux neurones de la couche n+1. Dans un réseau constitué de n couches, la première couche est reliée aux capteurs et la n^{ième} aux effecteurs. En d'autres termes, la première couche reçoit ses "entrées" du milieu extérieur au réseau et la couche de "sortie" fournit la réponse du réseau.

Les réseaux connexionnistes, possèdent une aptitude à l'apprentissage. L'apprentissage repose sur le renforcement des connexions entre les neurones concernés. On distingue trois types d'apprentissage :

- l'apprentissage dit supervisé dans lequel on présente simultanément au réseau une entrée et la sortie désirée de façon à pouvoir calculer la différence entre la sortie produite et la sortie désirée et minimiser l'erreur commise,

- l'apprentissage dit guidé ou renforcé dans lequel on indique uniquement au réseau si la sortie produite est correcte ou pas, ce qui entraîne une variation de poids des connexions,

- l'apprentissage non supervisé qui consiste à détecter de manière automatique les similitudes dans les exemples traités et à modifier les poids des connexions entre neurones pour que des exemples ayant les mêmes caractéristiques fournissent les mêmes sorties.

Enfin, le fonctionnement d'un réseau neuronal est composé de deux phases distinctes :

- une phase d'apprentissage pendant laquelle les poids des connexions sont mis à jour de manière dynamique ;

- une phase d'exécution pendant laquelle le réseau est effectivement opérationnel.

Nous ne distinguerons, dans ce domaine d'investigation, que les réseaux neuronaux adaptatifs et les cartes auto-organisatrices car ces deux modèles sont les plus proches de notre sujet.

3.6.1. Les réseaux adaptatifs non linéaires

Les réseaux adaptatifs non linéaires (RAN) [Renders, 1995] sont présents dans la majorité des systèmes biologiques dotés de capacités naturelles d'adaptation. Ces réseaux possèdent intrinsèquement deux niveaux : un niveau microscopique se comportant suivant des règles explicites de comportement et d'interaction des individus et un niveau macroscopique faisant apparaître des propriétés d'auto-organisation et d'émergence.

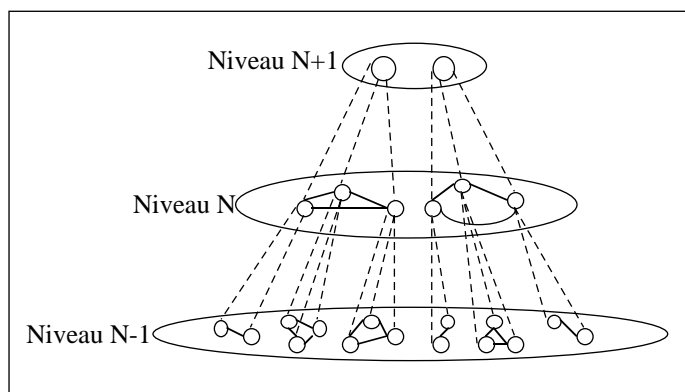


Figure 3.6.1. : Les réseaux adaptatifs non linéaires

On peut toutefois distinguer deux types d'auto-organisation dans les RAN : une auto-organisation dite verticale et une auto-organisation dite horizontale. Un système (bloc) à un niveau N est constitué d'un agencement de sous-systèmes (blocs) appartenant au niveau $N-1$. Réciproquement l'agrégation de certains sous-systèmes de niveau $N-1$ correspond à un des systèmes du niveau N . L'auto-organisation verticale consiste alors à rechercher en permanence une combinaison de sous-systèmes constituant de bons blocs au niveau supérieur. L'auto-organisation horizontale, quant à elle, s'exécute lors de la formation de bons blocs ou systèmes à un même niveau. Elle est mise en oeuvre suite aux rétroactions de l'environnement lors de compétitions d'agents de même niveau.

3.6.2. Les cartes auto-organisatrices

Les cartes auto-organisatrices s'inspirent de la modélisation des systèmes de perception chez les mammifères. Dans de tels systèmes, les signaux extérieurs sont réceptionnés puis traités à l'intérieur du système nerveux. Leur principale propriété est de pouvoir "coder" sur des neurones voisins des signaux d'entrée qui se "ressemblent".

Kohonen [Gurney, 1996], [Davalò, 1989], s'est inspiré de plusieurs observations biologiques et a eu l'idée de créer un réseau ayant les deux caractéristiques suivantes :

- il y a sélection du neurone qui correspond au type de signal d'entrée donné,
- il y a augmentation de l'activation de ce neurone et du groupe de neurones qui l'entoure lorsque ce signal d'entrée est présenté, ce qui revient à accroître la correspondance entre le signal d'entrée considéré ainsi que le neurone sélectionné et son voisinage.

La sélection consiste à trouver le neurone c tel que son vecteur de poids M_c soit le plus proche du vecteur d'entrée E et représente ainsi au mieux ce signal. L'augmentation de l'activation du neurone déterminé et du groupe de neurones qui l'entoure lorsque le signal est présenté nécessite la définition d'un voisinage V_c . La détermination V_c peut se faire de différentes façons ; les différentes simulations effectuées montrent qu'il vaut mieux choisir ce voisinage assez large au début de l'apprentissage pour le rétrécir par la suite. De même, la fonction d'oubli, proportionnelle à l'inactivité des neurones, doit, selon les simulations effectuées, être une fonction décroissante linéaire du temps d'apprentissage afin d'assurer l'aboutissement de celui-ci au bout d'un temps fini.

L'algorithme de Kohonen est donc le suivant pour chaque vecteur d'entrée :

- trouver le neurone k dont le vecteur poids est le plus près de l'entrée courante,
- modifier l'état de la cellule k et celui de toutes celles de son voisinage,
- diminuer le taux d'apprentissage lentement (afin d'ajuster finement les caractéristiques encodées),
- tous les M cycles, diminuer la taille du voisinage (afin de stabiliser le processus).

Les vecteurs de poids tendent alors à s'auto-organiser pour permettre de conserver sur la carte neuronale les relations de proximité des entrées. Une propriété intéressante des cartes auto-organisatrices est qu'elles reposent sur un modèle dont l'apprentissage est non supervisé : aucune sortie désirée n'est associée à chacun des vecteurs d'entrée.

Les modèles connexionnistes ont certes de grandes capacités d'auto-organisation mais leur structure (nombre de couches, connexions) nécessite d'avoir une vue complète de la tâche à exécuter. Ce besoin est loin d'être réaliste dans les systèmes sur lesquels nous désirons travailler.

3.7. L'AUTO-ORGANISATION DANS LES ALGORITHMES GÉNÉTIQUES

Considérablement utilisés dans le domaine de la vie artificielle [Renders, 1995], ces algorithmes permettent d'effectuer une recherche globale de manière itérative dans le but d'optimiser une fonction définie par l'utilisateur appelée fonction d'adéquation. L'algorithme travaille en parallèle sur une population d'individus ou chromosomes caractérisés par des gènes et possédant respectivement une fonction d'adéquation. Ces gènes sont le plus couramment représentés par des allèles égaux à 0 ou à 1. La finalité des algorithmes génétiques est donc de chercher la combinaison optimale de ces gènes afin d'obtenir l'adéquation maximale. Chaque itération de l'algorithme (génération) entraîne la création d'une nouvelle population comprenant le même nombre d'individus. Lors d'une nouvelle génération, les individus les moins adaptés sont remplacés par des "enfants" issus de la reproduction de "parents" choisis aléatoirement parmi les mieux adaptés. Pour cela, le système effectue des croisements et des mutations parmi les individus qui ont survécu. Au fur et à mesure des générations, les individus tendent vers l'optimum de la fonction d'adéquation : ceux qui perdurent après la procréation de nombreuses descendance sont ceux qui correspondent le mieux à la fonction d'adéquation.

Les algorithmes génétiques sont souvent utilisés dans les systèmes classifieurs apprenants. Ils sont appliqués, pour obtenir des règles pertinentes dans le classifieur. Les règles utilisées par ce dernier sont jugées suite à une rétroaction de l'environnement et un algorithme d'assignation de crédit appelé "bucket-brigade" (cf. § 2.3.3.). Le "bucket-brigade" est l'équivalent de l'algorithme de rétro propagation dans un réseau neuronal. A chaque règle est associée une valeur qui reflète la pertinence des règles du classifieur en fonction de l'expérience passée du système. L'algorithme du "bucket-brigade" est le mécanisme qui modifie cette valeur. Les algorithmes interprètent la force d'un classifieur comme le critère d'adaptation du système à un contexte donné et, périodiquement, ces algorithmes détruisent les règles ayant la force la plus faible et génèrent de nouvelles règles en modifiant celles ayant la force la plus importante par des opérations génétiques de croisement ou de mutation.

Outre l'opérateur de sélection, les opérateurs de croisement et de mutation forment la base d'un algorithme génétique. Leur but est de faire évoluer la population pour qu'elle converge vers un optimum global tout en effectuant une bonne exploration de l'espace de recherche.

3.7.1. L'opérateur de sélection

La **sélection** choisit les individus qui vont ensuite se reproduire. La méthode de sélection la plus simple consiste à tirer au hasard l'un d'entre eux (méthode de Monte-Carlo). Cette méthode ayant été jugée trop aveugle, des heuristiques ont été ajoutées afin d'améliorer l'efficacité de l'algorithme. Le principe de base a néanmoins été conservé. Parmi ces heuristiques on distingue le "crowding scheme" où seule une partie de la population est renouvelée, la stratégie élitiste consistant à conserver de manière intacte le meilleur individu dans la génération suivante ...

3.7.2. L'opérateur de croisement

Le **croisement** permet de concaténer un segment initial d'un agent i avec un segment final d'un agent j , en ayant au préalable déterminé la position du croisement c'est-à-dire la position de la cassure des segments respectifs de chaque agent. Si $a_i = \langle 0100100 \rangle$ et $a_j = \langle 1100110 \rangle$ et le croisement survient entre la troisième position et la quatrième position, alors $a_k = \langle 0100110 \rangle$ est le résultat du croisement. Une caractéristique de l'opération de croisement est qu'elle va produire une "progéniture" avec des spécialisations qui existent déjà dans la population courante. Par exemple, si chaque agent a un 0 en première position, l'opération de croisement ne pourra pas produire une progéniture avec un 1 à cette position. La seconde opération, la mutation, permet de résoudre ce problème.

3.7.3. L'opérateur de mutation

L'opérateur de **mutation** génère un nouvel agent en modifiant une ou plusieurs valeurs des gènes d'un agent existant en fonction de la position de mutation. Par exemple une mutation survenant dans l'agent $a_j = \langle 1101001 \rangle$ en cinquième position produirait $a_k = \langle 1101101 \rangle$. Une des caractéristiques de l'opération de mutation est qu'elle permet de garantir qu'une spécialisation d'un agent particulier ne sera jamais à 0 dans sa configuration.

Tout comme les modèles connexionnistes, la plupart des algorithmes génétiques nécessitent un contrôle centralisé et une connaissance parfaite de l'ensemble de la population. D'autre part, les

individus jouissent d'un degré d'autonomie quasiment nul. Ces caractéristiques vont à l'encontre des systèmes sur lesquels nous désirons travailler.

3.8. L'AUTO-ORGANISATION EN ROBOTIQUE

Cem Ünsal [Ünsal, 1993] et Luc Steels [Steels, 1990] se sont intéressés au phénomène d'auto-organisation dans le domaine de la robotique. Tous deux ont choisi la collecte d'objets comme finalité de leurs expériences et désirent établir un comportement coopératif entre agents. Les tâches de recherche et de collecte d'échantillons de roche sont réalisées par un groupe de robots autonomes qui doivent les porter ensuite à un endroit déterminé. Les agents n'ont qu'une vue partielle ; ils ne perçoivent qu'un nombre limité d'agents et d'objets. La difficulté est accrue chez Steels en raison d'un terrain parsemé d'obstacles, de collines, de vallées... Les agents sont individuellement incapables de transporter la charge. Chez Ünsal, les agents sont obligés de se regrouper pour pouvoir faire office de transporteur. Chez Steels, la charge est fractionnable, ce qui permet aux agents de transporter des fragments de charge. Ils doivent cependant, dans ce dernier cas, faire en sorte d'optimiser leur déplacement. Les échantillons étant groupés en certains points, les robots à force de coopération, vont pouvoir accomplir leur tâche.

Leur problématique peut se résumer de la manière suivante : comment des robots autonomes et distribués géographiquement peuvent se coordonner pour trouver des échantillons et les transporter jusqu'à un endroit déterminé ?

3.8.1. Auto-organisation par auto-catalyse

La stratégie adoptée par Luc Steels est la suivante : au début de l'expérience, les agents errent de manière aléatoire jusqu'à trouver une ressource. Lorsqu'un robot a découvert un échantillon, il doit communiquer sa découverte aux autres robots et établir un chemin efficace entre la localisation de l'échantillon et celle du dépôt. La communication entre robots s'effectue par le biais de dépôts de miettes sur le sol. Chaque robot possède donc un certain stock de miettes et est à la fois capable de les déposer par terre et de les ramasser.

Afin de pouvoir mettre en oeuvre cette stratégie, les comportements suivants sont ajoutés au répertoire des comportements des robots mobiles :

- 1 - Si je porte un échantillon, je laisse tomber deux miettes,
- 2 - Si je ne porte aucun échantillon et que des miettes sont détectées, je prends une miette.

La règle 1 va établir le chemin ; elle permet de mettre en oeuvre le phénomène d'auto-catalyse. La règle 2 va démolir le chemin.

Il a également ajouté une troisième règle au répertoire des comportements des robots : si je ne porte pas d'échantillon et si je détecte des miettes, je me déplace vers la plus grosse concentration de miettes. Les autres règles de déplacement reposent sur les mouvements aléatoires, les mouvements de retour, les mouvements d'exploration et les contournements d'obstacles.

Le travail de Luc Steels est très proche de celui effectué sur les sociétés animales en particulier chez les fourmis en quête de nourriture qui laissent des traces de phéromone lorsqu'elles retournent d'une source de nourriture vers leur nid. Contrairement aux travaux concernant le fourragement où les fourmis tirent profit du phénomène de chaos pour découvrir de nouvelles ressources, la troisième règle empêche ici l'apparition de fluctuations aléatoires.

3.8.2. Auto-organisation spatiale par attraction/répulsion

Le travail de Cem Ünsal s'intéresse à la manière dont des robots peuvent dynamiquement circonscrire une charge pour pouvoir la soulever et la transporter le plus efficacement possible par la suite, car individuellement ils ne peuvent pas y parvenir.

Cem assimile les objets devant être collectés (charges) à des signaux lumineux perceptibles par l'ensemble des robots. Chaque robot n'a qu'une vue locale et a la faculté de détecter et de différencier les signaux en provenance d'autres agents et ceux émis par les objets. Il doit également être capable de calculer la distance ainsi que la direction de la source du signal.

Les algorithmes proposés reposent sur les notions d'attraction et de répulsion. En fonction de leur position par rapport à la charge convoitée, les agents s'éloignent ou se rapprochent de celle-ci ou encore s'écartent de leur congénère en fonction d'un champ de gravitation paramétrable. Les agents, en fonction de règles très simples, entourent l'objet désiré en formant des figures géométriques particulières (cercles, paraboles) ; ils n'ont pas conscience de leur disposition spatiale, leur seul but est de se rapprocher efficacement de la charge. Pourtant, la forme géométrique obtenue et résultant de leur déplacement donne lieu à une forme qui peut être vue comme le résultat d'un comportement collectif.

Le système proposé est également adaptatif dans le sens où lorsqu'un agent tombe en panne, ou lorsqu'un signal bouge, la définition du champ de gravitation change et les agents se déplacent alors vers leur nouvelle position.

Contrairement à ce que nous souhaitons, Ünsal fait un présupposé sur la finalité du système : les agents s'auto-organisent pour circonscrire efficacement une charge de manière à pouvoir la porter.

3.9. L'AUTO-ORGANISATION EN INTELLIGENCE ARTIFICIELLE DISTRIBUÉE

Peu de travaux issus du domaine de l'intelligence artificielle distribuée traitent de l'auto-organisation. Nous allons étudier quatre articles se rapportant à cette problématique.

3.9.1. Auto-organisation par décomposition/recomposition

Les Gasser et Toru Ishida [Gasser, 1991] insistent sur le caractère essentiel d'une correspondance entre l'architecture d'un système de résolution de problème distribuée, la structure des problèmes qu'il résout, et les conditions environnementales dans lesquelles il les résout. Dans un monde dynamique peuplé d'agents, cette correspondance doit être maintenue par une adaptation dynamique.

Ils ont, pour cela, défini une valeur R indiquant le rendement de chaque agent et donné la possibilité à un agent de savoir si l'organisation courante atteint le critère de performance $PERF_{désirée}$ par comparaison avec $PERF_{actuelle}$, performance mesurée le plus récemment. Lorsque $PERF_{actuelle} > PERF_{désirée}$ la performance de l'organisation doit être améliorée. Dans le cas inverse, l'organisation a les moyens de libérer des ressources.

Enfin, ils ont établi des règles de réorganisation afin de déclencher la réorganisation au moment opportun. Nous pouvons résumer ces règles de la manière suivante : les agents se décomposent eux-mêmes lors d'une surcharge de travail afin d'accroître le parallélisme et composent avec un autre agent pour libérer des ressources de matériel. Chaque décision de réorganisation invoque une des deux primitives de réorganisation :

- la **décomposition** qui casse la correspondance entre une collection de règles de résolution de problème et leur interpréteur, en créant un second interpréteur (un agent) pour quelques règles, en insérant les actions de communication entre les agents, et en appliquant la synchronisation entre des actions de résolution de problème,

- la **composition** qui fusionne deux voisins indépendants (agents avec des actions interdépendantes). Elle crée une relation "fermée" sur les groupes de connaissances de résolution de problème, supprime les messages inter-agent et libère les ressources de calcul et de communication.

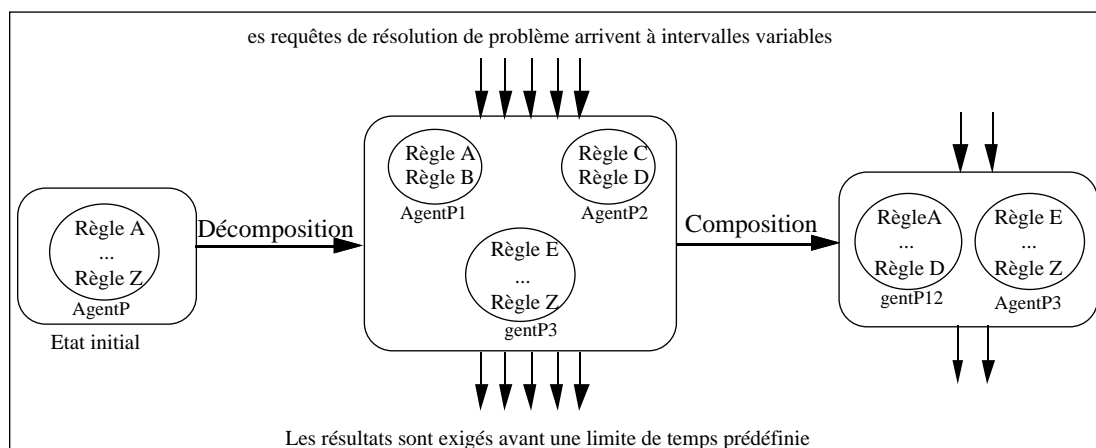


Figure 3.9.1. : Illustration des primitives de composition et de décomposition

Frédéric Guichard, dans sa thèse [Guichard, 1996], propose une méthode d'auto-organisation basée sur deux types de réorganisation : la réorganisation des compétences des agents et la réorganisation physique qui permet une utilisation des ressources physiques du système proche de l'optimum.

La réorganisation des compétences est basée sur des primitives de composition et de décomposition équivalentes à celles définies par Gasser et Ishida. Ces deux primitives font partie de la compétence des agents. Deux autres compétences ont été définies :

- l'**introspection** cognitive : elle permet à l'agent d'avoir un point de vue sur sa charge de travail, sur la fréquence d'utilisation de ses compétences et la durée d'exécution de leurs méthodes. Elle permet également d'évaluer le temps d'exécution des méthodes des autres agents pour toutes les compétences connues par l'agent.

- la **suppression** : la suppression survient lorsqu'un agent a une charge de travail trop peu conséquente pour justifier de son existence. L'agent va alors prendre la décision de s'auto-détruire, en prenant soin auparavant d'avertir de sa disparition les agents qu'il connaît ainsi que son agent système et en ayant pris soin de transférer ses compétences.

La réorganisation physique s'effectue, quant à elle, en trois étapes consistant respectivement en :

- la détermination des conditions locales rendant le placement ou la migration d'un agent souhaitable,

- la détermination du site sur lequel l'agent doit être transféré en calculant les forces gravitationnelles existantes entre l'agent courant et les sites qu'il connaît. Ce calcul est fonction des volumes de messages échangés et des caractéristiques du lien de communication (c'est-à-dire de la distance qui sépare l'agent des sites voisins).

- la spécification de la nature et de la quantité d'informations utilisées pour la prise de décisions.

Dans les deux travaux proposés on peut remarquer qu'il est impossible de modifier les liens entre agents sans que ces derniers ne soient ou décomposés ou composés avec d'autres. Les agents pourraient pourtant supprimer des liens jugés inutiles [Foisel, 1996] sans toutefois modifier leurs compétences en se décomposant ou en se composant.

D'autre part, ces deux primitives peuvent être dangereuses dans le sens où une incohérence dans les connaissances d'un agent peut perdurer et même se propager s'il se duplique.

Les travaux de Gasser ont un désavantage qui a été corrigé par la primitive d'introspection rajoutée par Guichard : les primitives de décomposition et de composition n'étaient pas amorcées par les agents eux-mêmes mais par une valeur globale (l'état de l'organisation à un instant donné).

Enfin, une dernière critique adressée aux deux travaux : l'auto-organisation n'est déclenchée que pour accroître le parallélisme et pour distribuer les agents le plus efficacement possible, ce afin de rendre le système plus performant. La réception d'une perturbation en provenance de l'environnement n'a pas été prise en compte dans les modèles décrits.

3.9.2. Auto-organisation par systèmes multi-agents récursifs

Pierre Marcenac [Marcenac, 1997a] propose une architecture permettant le développement d'applications simulant des systèmes complexes dans le but d'étudier la dynamique du système et de comprendre l'émergence de comportements.

Il propose une architecture récursive de systèmes multi-agents comportant trois niveaux.

Le niveau le plus haut (niveau 1), appelé "Système Multi-Agents" décrit le système dans sa globalité. Il est composé d'agents de niveau intermédiaire.

Le niveau intermédiaire (niveau 2), appelé "Agents cognitifs", décrit la composition d'un agent du premier niveau. Son rôle est de modéliser la dynamique de l'adaptation et de reproduire des comportements de plus haut niveau d'abstraction que ceux des agents réactifs qui le constituent.

Le niveau le plus bas (niveau 3), appelé "Agents réactifs", décrit les comportements linéaires des composants d'un agent du second niveau.

Ainsi, un agent peut toujours être perçu comme un système multi-agent aux niveaux 1 et 2, alors qu'un système multi-agent peut toujours être perçu comme un agent aux niveaux 2 et 3.

Les interactions entre agents de même niveau sont mises en oeuvre par envoi de messages asynchrones. Les interactions entre agents de niveaux adjacents sont mises en oeuvre par deux mécanismes : la recomposition et la décomposition. La recomposition est employée lorsqu'une instabilité est détectée à un niveau donné. Elle doit alors être signalée à la société (donc au niveau supérieur). La décomposition est quant à elle utilisée pour transférer des informations d'un niveau donné au niveau immédiatement inférieur.

L'auto-organisation repose sur la "coopération" entre agents [Marcenac, 1997b]. L'état d'un agent est représenté par un vecteur d'état P dans lequel chaque coordonnée décrit une propriété interne de l'agent. Le mécanisme déclencheur de l'auto-organisation consiste à détecter une similarité entre vecteurs d'états d'agents de granularité identique qui sont dans un état instable en raison d'une perturbation extérieure. La similarité est basée sur la comparaison des vecteurs d'état de l'agent courant avec ceux de ses voisins. Si le vecteur est similaire à celui d'un de ses voisins, les deux agents vont être regroupés et inscrits dans une structure auto-organisée par une fonction méta. La fonction méta permet de créer un agent intermédiaire pour agréger des agents de fine granularité similaires, ou pour ajouter de tels agents dans un agent existant. Une structure intermédiaire va alors émerger, résultat de l'instabilité du niveau adjacent inférieur et responsable de l'évolution des agents de granularité inférieure qui la constituent.

L'architecture proposée a été instanciée pour simuler les éruptions volcaniques.

Comme le souligne l'auteur, seule la similarité est considérée alors que la prise en compte de la complémentarité ou de l'antagonisme pourrait être plus adéquate lors de simulations de sociétés artificielles.

3.9.3. Auto-organisation par rétroaction

Luc Steels [Steels, 1996a], [Steels, 1996b], propose un ensemble de mécanismes grâce auquel un groupe d'agents distribués peut développer de manière autonome un langage en conversant avec d'autres agents présents dans l'environnement. Chaque agent possède son propre ensemble d'associations mot/signification et peut expérimenter les associations utilisées par d'autres agents présents dans le système.

Pour un agent particulier et une signification particulière, trois possibilités et actions peuvent survenir :

1 - l'agent possède un mot pour la signification. Cette association changera en fonction de la probabilité qui dépend du succès de l'association.

2 - l'agent ne possède pas de mot pour cette signification. Dans ce cas l'agent sélectionne un mot à partir des mots contenus dans la conversation et l'associe à la signification. S'il n'y a pas suffisamment de mots libres, l'agent peut créer un nouveau mot et alors créer une nouvelle association.

3 - l'agent ne connaît pas la signification. Dans ce cas la signification est adoptée et un mot est choisi pour elle, en utilisant le même principe que celui décrit dans le cas précédent.

Trois mécanismes sont donc mis en oeuvre :

- la propagation, lorsque des agents adoptent des associations mot/signification possédées par d'autres agents,
- la création, lorsqu'un agent génère un nouveau mot et l'associe avec un ensemble de caractéristiques découvertes,
- une auto-organisation due à un mécanisme de rétroaction positive entre association, sélection et succès dans l'utilisation.

La cohérence est atteinte au travers de l'auto-organisation par la formation spontanée de structures au travers de la mise en application de variations aléatoires (fluctuations). Ces fluctuations sont ici causées par différentes associations circulant dans la population. Un agent mémorise combien de fois une association mot/signification a été utilisée dans le jeu dans lequel il est impliqué et combien de fois elle a été couronnée de succès. Lorsque des significations ont besoin d'être codées, l'agent choisit l'association la plus communément utilisée avec succès. Ceci introduit une boucle de rétroaction positive : plus un mot est utilisé à bon escient plus l'association sera couronnée de succès. Lorsqu'il y a plusieurs possibilités, une lutte temporaire s'engage jusqu'à ce qu'une association gagne.

Dans une telle méthode d'auto-organisation, les agents ne jouissent pas d'une grande autonomie. Ils possèdent chacun un objectif, qui est finalement celui de la collectivité : celui de tester les associations mot/signification qu'il possède et d'apprendre en fonction de ce qu'il reçoit des autres. Cette hypothèse est contraire à ce que nous souhaitons, à savoir une méthode d'auto-organisation indépendante de la fonction globale du système.

3.9.4. Auto-organisation par induction

Le travail de Bollen [Bollen, 1996], [Bollen, 1995] consiste à développer des algorithmes permettant au WWW de changer de manière autonome sa structure et d'organiser la connaissance qu'il contient en "apprenant" les idées et la connaissance des usagers humains en fonction du comportement de ces derniers lors d'une recherche. En d'autres termes, il désire concevoir un algorithme permettant aux réseaux hypertextes distribués de s'auto-organiser selon la connaissance de leurs usagers de sorte à rendre toute recherche plus performante. Bollen propose d'utiliser des métaphores et d'implémenter les conditions nécessaires pour faire du WWW une fonction plus proche d'un cerveau.

Il propose pour cela un réseau hypertexte dans lequel chaque connexion entre deux noeuds A et B est associée à une mesure unidirectionnelle de la force de leur relation associative. La force de ces

relations est dynamiquement dérivée des modèles de navigation des usagers grâce à un ensemble de 3 règles d'apprentissage. Ces règles sont les suivantes :

- la fréquence : si la connexion d'un noeud A vers un noeud B est utilisée, elle va être renforcée de F_b . Si S exprime la force de la connexion entre deux noeuds A et B qui sont consécutivement consultés par un usager, on a alors $(A \rightarrow B) \Rightarrow S(A,B) + F_b$, $F_b = 1$.

- la symétrie : si la connexion d'un noeud A vers le noeud B est utilisée, la connexion du noeud B vers le noeud A va être renforcée d'un bonus S_b . On a donc $(A \rightarrow B) \Rightarrow S(B, A) + S_b$, $S_b = 0,3$.

- la transitivité : si la connexion d'un noeud A vers le noeud B est utilisée, et ensuite la connexion du noeud B vers le noeud C est utilisée, la connexion du noeud A vers le noeud C est renforcée d'un bonus T_b . On a donc $(A \rightarrow B) \& (B \rightarrow C) \Rightarrow S(A,C) + T_b$, $T_b = 0,5$.

Seules les deux dernières règles d'apprentissage permettent la création de nouveaux liens. Les résultats de ses expériences ont montré que les réseaux expérimentaux de noms s'auto-organisent selon ces trois règles d'apprentissage jusqu'à atteindre un état stable qui représente les structures des associations et structures de connaissances partagées entre participants.

Bollen a donc montré que des règles d'apprentissage simples peuvent être suffisantes pour permettre à un réseau associatif de s'auto-organiser jusqu'à ce qu'il reflète plus ou moins les sémantiques intuitives de ses usagers.

La méthode préconisée n'a été pour l'instant appliquée que sur des réseaux de mots dont la complexité est très loin de celle des documents HTML contenus dans le WWW. D'autre part, elle ne tient pas compte d'éventuelles perturbations en provenance de l'environnement puisque l'éventail des possibilités présentées à l'utilisateur est borné.

3.10. ÉMERGENCE

Bien que le mot "émergence" n'ait pas été fréquemment énoncé durant cet exposé sur l'auto-organisation, on peut percevoir les corrélations qu'il existe entre ces deux notions. Au niveau de l'être humain, parfait système auto-organisateur, l'émergence se traduit par exemple chez le jeune enfant par sa capacité à parler. Avant de faire un rapprochement entre ces deux concepts, nous allons donner un très bref aperçu des travaux traitant de l'émergence.

La notion d'émergence s'oppose à l'approche dite "classique" pour aborder la résolution d'un problème particulier. Elle ne consiste pas à décomposer un problème en sous-problèmes puis à résoudre individuellement et consécutivement chacun d'eux pour pouvoir atteindre la solution cherchée mais à doter les agents de comportements qui les amèneront à interagir de manière locale pour résoudre le problème donné. De ces interactions émergera peu à peu une organisation stable et structurée des relations entre agents. Il s'agit d'émergence dans le sens où la fonctionnalité globale

du système n'est pas explicitement écrite dans le comportement des agents. Un comportement émergent est donc caractérisé par une dynamique globale, supérieure à la simple somme des interactions des constituants du système. L'émergence fait implicitement apparaître au moins deux niveaux de description de la complexité du système : un niveau "micro" constitué d'éléments de faible granularité en interaction ainsi qu'un niveau "macro" constitué d'éléments de plus grosse granularité dans lequel apparaissent les propriétés émergentes. Les comportements du micro-niveau sont généralement simples, linéaires et prédictibles alors que ceux du niveau macro sont la plupart du temps non linéaires. Le lien causal existant entre le niveau micro et le niveau macro est très difficilement explicable d'où le qualificatif d'émergent.

Alexis Drogoul [Drogoul, 1993] différencie la notion d'émergence ou de "fonctionnalités qualitatives" à celle de "fonctionnalités amplifiées". Cette dernière fait référence à des entités résolvant une tâche donnée de manière similaire qu'elles soient seules ou au contraire plusieurs. Le phénomène alors observé correspond à un saut quantitatif caractéristique d'un effet de groupe. Cette distinction repose sur le fait que l'émergence nécessite une quantité d'individus minimale pour atteindre un objectif alors que ce besoin n'est pas nécessaire dans le second cas. D'autre part, les phénomènes collectifs peuvent être expliqués dans le cas des fonctionnalités qualitatives alors que ce n'est pas le cas lorsqu'il y a émergence.

Stéphanie Forrest [Forrest, 1990] différencie les notions d'émergence et d'épiphénomène. Selon elle, les phénomènes émergents observés ne sont pas les modèles globaux qui apparaissent au macro-niveau mais l'interprétation qui peut être faite. Comme l'illustre [M.R. JEAN, 1997], le phénomène émergent chez les fourmis n'est pas, dans ce cas là, la trace de phéromones qui s'est peu à peu construite mais l'interprétation qu'en fait l'observateur, c'est-à-dire la formation d'un chemin entre un nid et une source de nourriture.

L'émergence est différemment perçue : cela vient très certainement du caractère incertain qui lui est associé. En effet, étant donné que la fonctionnalité émergente d'un système n'est codée dans aucun des comportements des agents, un risque subsiste quant à sa pertinence : elle peut correspondre à un résultat escompté tout comme elle peut fournir un résultat aberrant.

Les notions d'émergence et d'auto-organisation sont fortement liées. L'auto-organisation introduit des comportements, des structures ou des formes au niveau collectif qui sont nouveaux (nouvelles) relativement aux comportements, structures ou formes des parties qui le composent : c'est l'émergence. L'auto-organisation telle que nous la concevons ne fait aucun pré-supposé quant à la finalité du système. Les entités concevant le système auto-organisateur doivent seulement poursuivre un objectif individuel tout en s'adaptant aux éventuelles perturbations extérieures et en apprenant au fur et à mesure de leur fonctionnement. Bien qu'aucune finalité ne soit explicitement programmée dans le comportement des agents, le système auto-organisateur parvient à fournir un résultat. C'est en cela qu'il y a émergence.

3.11. DISCUSSION

L'auto-organisation correspond à notre sens à la capacité d'une société d'agents d'apprendre de manière autonome et de s'adapter à tous signaux perçus par l'environnement ou échangés entre entités qui constituent cette société. Les entités qui composent le système ne poursuivent qu'un objectif individuel et ne connaissent rien de la fonction globale devant être produite par le système. Elles profitent du signal perçu de l'environnement pour enrichir les liens d'interaction qui les unissent les unes avec les autres.

Notre définition est conforme à celle proposée par Jean-Claude Perez, [Perez, 1988]. Il différencie les notions d'adaptation, d'apprentissage et d'auto-organisation à l'aide des questions/réponses. Au préalable, il fait l'hypothèse qu'il se positionne dans un contexte où un système en équilibre interne est soumis à une perturbation extérieure. Son raisonnement est alors le suivant :

“① - Le système peut-il assimiler cette perturbation et atteindre un nouvel état d'équilibre ?

Si tel est le cas, le système aura la faculté d'auto-adaptation

② - Le système pourra-t-il, plus tard, reconnaître le même signal, s'il se reproduit ?

Si le système sait répondre “oui” aux deux premières questions il sera doté d'auto-apprentissage

③ - Le système exploitera-t-il cette perturbation pour accroître son niveau global d'organisation et de complexité ?

Si le système sait répondre “oui” au trois questions, il a des qualités d'auto-organisation.”

Peu de travaux parmi ceux étudiés ne font pas de présupposé sur la finalité du système. Dans les systèmes naturels, les entités qui composent le système poursuivent individuellement un objectif propre et ne connaissent en aucun cas le travail qu'elles effectuent de manière collective : seul un observateur extérieur pourra voir émerger le résultat du travail collectif. Ces sociétés ont la faculté de s'adapter aux changements de leur environnement et d'apprendre en conséquence. Ce sont de véritables phénomènes auto-organisateurs. Les travaux de Steels sur les robots fourrageurs possèdent également cette propriété puisqu'ils calquent les comportements des insectes dans les systèmes naturels (seuls les comportements chaotiques observables dans les systèmes naturels n'ont pas été considérés). A un niveau de complexité bien moindre (par le contexte d'expérimentation exposé dans les articles étudiés) les travaux de Bollen sur la réorganisation dynamique d'un réseau hypertexte et ceux de Steels sur les vocabulaires auto-organisateurs sont également indépendants de la finalité du système. Dans tous les autres travaux, la mise en oeuvre de l'auto-organisation n'est pas, à notre sens, suffisamment indépendante du contexte. Pour permettre aux systèmes de s'auto-organiser face à l'imprévu, le système doit être indépendant de toute représentation du monde : les mécanismes auto-organisateurs doivent être non cognitivistes.

D'un point de vue bien plus pragmatique, nous pouvons noter que les travaux présentés dans cet état de l'art sur l'auto-organisation n'appartiennent pas aux mêmes domaines d'investigation et, malgré cet état de faits, des similitudes existent sur la mise en oeuvre de l'auto-organisation. Les mécanismes les plus couramment employés reposent sur l'utilisation de boucles de rétroaction positive ou négative. Un certain ordre émerge alors du système, appelé "ordre par fluctuation". Lorsque le système est loin de l'équilibre, une petite fluctuation peut, dans certaines conditions favorables, s'amplifier jusqu'à "unifier" et auto-organiser la totalité du système. On arrive alors à un comportement "holiste". Dans les sociétés d'insectes par exemple, le marquage phéromonal est à l'origine de ces fluctuations. L'auto-organisation est également mise en oeuvre par décomposition et recombinaison d'agents [Gasser, 1991], [Guichard, 1996] voire même par récursivité de systèmes multi-agents [Marcenac, 1996].

Un reproche que nous pouvons faire à l'encontre de la quasi totalité de ces travaux (exceptés ceux de Pierre Marcenac) est qu'aucun comportement n'a été spécifié dans le cas où le système est soumis à une perturbation extérieure. La mise en oeuvre de l'auto-organisation est dans la plupart des cas fonction de l'adéquation de la solution produite par le système par rapport au résultat escompté. Pourtant, la réelle puissance d'un système auto-organisateur repose sur ses capacités d'adaptation aux perturbations pouvant survenir en cours de son histoire, ceci, sans préjuger des résultats escomptés.

Enfin nous terminerons par une citation de Stephan Kirn [Kirn, 1996] visant à encourager l'étude de l'auto-organisation : "l'intelligence artificielle distribuée fait une distinction entre l'apprentissage organisationnel à court terme et celui à long terme. Le premier concerne la flexibilité organisationnelle alors que le second s'intéresse à la modification des structures organisationnelles comme résultat des changements dans l'environnement [Gasser, 1992]. Les approches les plus importantes concernent l'auto-organisation des compétences des systèmes multi-agents [...] les chercheurs se sont tout d'abord intéressés à la manière dont ces systèmes sont capables de se réorganiser eux mêmes de sorte à faire face aux problèmes changeants de l'environnement dynamique. Cependant, personne du domaine de l'IAD ne s'intéresse à l'apprentissage organisationnel de la sorte. Ceci est très surprenant parce qu'un nombre important d'applications en IAD (processus de décision, coopération informatique d'humains, ...) utilise grandement l'apprentissage de l'organisation." Cette remarque nous conforte à la fois sur l'utilité et la pertinence de l'apprentissage organisationnel en univers multi-agent.

CHAPITRE 4

PROPRIÉTÉS DES SYSTÈMES MULTI-AGENTS COOPÉRATIFS

4.1. INTRODUCTION

Nous pensons qu'il est possible de résoudre des problèmes dans lesquels la dynamique est importante et où aucun agent ne possède une vue globale du système ni même une idée sur la finalité du système. La solution doit se construire dynamiquement au fur et à mesure des interactions entre les entités composant le système ainsi que des interactions entre le système et son environnement (ce dernier statuant en dernier ressort sur la fonction globale du système).

Notre but est de construire un système multi-agent fonctionnellement adéquat. Un système est qualifié de fonctionnellement adéquat s'il réalise correctement la fonction pour laquelle il a été conçu, indépendamment de la dynamique de l'environnement dans lequel il évolue. L'adéquation fonctionnelle doit être constante et garantie durant la vie du système. Cette constance signifie que les perturbations qui surviendront durant son fonctionnement seront assimilées au moment de leur apparition par le système.

Nous proposons une méthode suffisamment générique ; elle ne fait aucun présupposé sur la finalité du système. Il en est de même pour tous les agents qui constituent le système. Seul un observateur de l'application est à même de pouvoir statuer sur la fonction globale réalisée par le système.

Ce chapitre se compose de cinq paragraphes : un premier paragraphe décrivant la propriété sur laquelle repose notre méthode d'apprentissage, ainsi que les hypothèses et notations adoptées. Les trois paragraphes suivants concernent la démonstration de la propriété proposée. Enfin, le dernier paragraphe explicite la manière de concevoir et spécifier un problème pour obtenir un système fonctionnellement adéquat conformément aux propriétés démontrées.

4.2. CADRE D'ÉTUDE

Nous allons introduire dans ce paragraphe la propriété sur laquelle repose notre méthode d'apprentissage et les hypothèses utiles pour la démonstration de celle-ci. Nous présenterons ensuite le contexte dans lequel nous nous plaçons et les notations adoptées. Nous terminerons enfin en indiquant les différentes étapes de la démonstration qui feront l'objet des trois paragraphes suivants.

4.2.1. Propriété à démontrer

La propriété que nous énonçons repose sur la métaphore suivante : un logiciel écrit par un programmeur, ne fournira les résultats attendus que s'il a été bien conçu. "Bien conçu" signifie que les divers appels de sous-programmes sont effectués au bon endroit et au bon moment. Les paramètres d'entrée d'un sous programme doivent permettre à ce dernier d'effectuer le travail pour lequel il a été spécifié, c'est-à-dire lui donner les informations sur lesquelles il va pouvoir raisonner et grâce auxquelles il va produire un ou plusieurs résultat(s). Ce ou ces résultat(s) devra ou devront être transmis à leur tour comme paramètre d'entrée du prochain sous-programme. Ces divers appels de sous-programmes représentent l'organisation du programme général et les divers sous-programmes représentent les différentes parties du programme général qui n'est autre qu'un système à part entière.

L'enchaînement qui aboutit à une bonne solution représente intuitivement une situation coopérative. Autrement dit nous pouvons dire que si les différentes parties d'un système entretiennent des relations coopératives (dans le sens où elles interagissent sans s'induire en erreur mais au contraire en s'aidant les unes les autres) le système fournit un résultat correct. Nous faisons bien évidemment le présupposé, dans le cas présent, que les diverses parties réalisent bien ce pour quoi elles ont été conçues à partir du moment où les informations nécessaires à leur raisonnement leur sont fournies.

La propriété que nous énonçons et que nous allons démontrer dans un cadre très général est la suivante :

“Pour tout système fonctionnellement adéquat dans un environnement donné, il existe au moins un système à milieu intérieur coopératif qui réalise une fonction équivalente”.

4.2.2. Hypothèses

Hypothèse 1 : Les hypothèses de travail pour notre méthode d'apprentissage sont cohérentes avec les propriétés des systèmes multi-agents. Ce sont celles présentées dans le chapitre de l'état de l'art consacré à l'apprentissage dans les systèmes multi-agents. Un système multi-agent doit favoriser l'aspect distribué et être ouvert. De plus, la conception d'un système multi-agent doit conserver l'autonomie des agents et le principe de localité. Un des points fondamentaux est la coopération entre les agents. C'est pourquoi notre méthode d'apprentissage doit respecter et étendre ces principes.

De plus, les agents sur lesquels nous travaillons sont composés de compétences, de croyances sur les autres et d'attitudes sociales. D'une manière générale les attitudes sociales peuvent être définies à différents degrés : la sincérité ou le mensonge, l'altruisme ou l'égoïsme... Elles conditionnent fortement le comportement d'un individu au sein d'un collectif.

Hypothèse 2 : Notre méthode d'apprentissage suppose que toutes les parties nécessaires à la réalisation d'un système fonctionnellement adéquat sont présentes dans le système. Cette hypothèse peut paraître très restrictive, c'est pourquoi nous y répondrons après la démonstration, au paragraphe 5.1.

4.2.3. Contexte

Un système est immergé dans un environnement qu'il ne perçoit que partiellement et tous deux s'influencent mutuellement ; il est donc possible de parler d'activité d'un système ou d'un environnement. Un état particulier de l'environnement peut induire une transformation du système (qui devient alors actif) dont le résultat transforme à son tour l'environnement (qui devient actif) ... et ce jusqu'à ce qu'ils s'ajustent mutuellement.

Pour illustrer ce type d'interactions considérons deux personnes en désaccord sur un sujet et qui tentent de se convaincre. La personne A (considérons la comme le système) va agir dans son environnement (constitué de la personne B) en argumentant son point de vue. Son action sur le monde transforme l'environnement (la personne B) qui va lui-même agir en confirmant ou en infirmant les propos tenus par A. L'action de l'environnement amènera la personne A à agir à son tour dans le monde, etc. ce jusqu'à atteindre un compromis entre les deux protagonistes.

L'environnement dans lequel est immergé le système exerce une contrainte sur le système à laquelle ce dernier doit s'adapter et vice versa.

La dynamique d'un système provient de l'itération entre l'interprétation par les agents de leur environnement local, les actions des agents sur cet environnement, la nouvelle interprétation de l'environnement modifié, les nouvelles actions, etc. Quand une telle dynamique (ou certain de ses composants) se stabilise (attracteur) on peut parler d'émergence d'une structure ou d'une fonctionnalité globale [Lenay, 1996].

Les interactions entre un système et son environnement sont quasi permanentes. De manière intuitive, nous avons distingué trois types d'activités induites par ces interactions :

- l'activité **coopérative**. La transformation du monde effectuée par le système ou l'environnement favorise l'activité de l'autre : ils ont des activités dites complémentaires. Un exemple de ce type d'interactions se retrouve chez les roussettes et les arbres fruitiers dont elles mangent les fruits, sans distinguer la pulpe des graines. Les graines non assimilées sont rejetées dans les excréments de l'animal lorsqu'il est en vol. Les arbres fruitiers, incapables de proliférer seuls étant donnée la zone beaucoup trop ombragée dans laquelle ils vivent, le font par le biais de ces animaux, qui eux aussi en retirent profit puisque cela leur permet de se nourrir.
- l'activité **antinomique**. La transformation du monde accomplie par le système ou l'environnement est contradictoire avec un état du monde nécessaire pour la poursuite des activités de l'autre. Un exemple typique est le partage de ressource : elle peut être accaparée par un organisme au détriment d'autrui.
- l'activité **indifférente**. La transformation du monde effectuée par le système ou l'environnement ne favorise pas et n'empêche pas l'activité de l'autre.

Cette classification se retrouve en partie chez Ferber [Ferber, 1995] dans la définition qu'il donne de l'interaction avec des buts compatibles et incompatibles. Il en arrive à définir des situations d'indifférence et de coopération, conformes à la définition que nous en donnons. Julia Rose Galliers [Bouron, 1992] a fait cette même classification pour définir trois postures propositionnelles qui peuvent connoter un but : le conflit, la coopération et l'indifférence. De même, Susanne Kalenka [Kalenka, 1997] note que "à la fois l'agent qui réalise une action et la société dans laquelle l'agent est situé peuvent être affectés par son exécution. Cet effet peut être bénéfique (une valeur d'utilité positive), préjudiciable (une valeur d'utilité négative) ou indifférent (une valeur d'utilité nulle)."

Dans le monde réel, toutes les relations que peut entretenir un système avec son environnement ne peuvent être cataloguées dans une seule de ces catégories : elles changent de nature au cours du temps. Plus précisément, à chaque instant, une unique activité peut avoir des conséquences simultanées différentes. Il est donc possible de dire qu'un état du monde résultant de l'activité du système ou de l'environnement peut être décomposé en trois parties complémentaires selon leurs

conséquences pour l'autre : une partie coopérative, une partie antinomique et une partie indifférente (une voire même deux de ces parties pouvant être vide(s)).

4.2.4. Notations

Les notations adoptées pour la formalisation et la démonstration de la propriété énoncée sont explicitées dans ce paragraphe.

Soit W le monde dans lequel nous nous plaçons.

Soit S un système du monde W .

Soit E l'environnement dans lequel S est immergé.

W est composé à la fois de S et de E .

Soit f_s la fonction que réalise le système S . La fonction que réalise un système dépend des supports physiques mis en jeu. C'est en l'occurrence le cas lorsque les systèmes sont jugés sur leur rapidité de réponse. La vitesse de traitement du système entre alors en ligne de compte pour le jugement de l'adéquation du système par rapport à la fonction pour laquelle il a été conçu. La fonction réalisée par le système ne peut donc être dissociée des supports physiques mis en jeu pour sa réalisation.

\mathcal{R}_w est l'ensemble de tous les états du monde. $W_i \in \mathcal{R}_w$ ($i \in \mathbb{N}$) est un état particulier du monde W . Le système S ne perçoit et n'agit que localement sur l'environnement dans lequel il est immergé. Soit $W_\alpha^S \in \mathcal{R}_w$ l'état du monde du point de vue du système S . L'activité f_s réalisée par le système S peut être décrite de la manière suivante : $\exists W_\beta^S \in \mathcal{R}_w \mid f_s(W_\alpha^S) = W_\beta^S$.

Dans ce cas, W_α^S représente l'état courant considéré par S pour effectuer une action et W_β^S désigne le nouvel état du monde une fois que S a réalisé son action, cet état du monde étant celui perçu par l'environnement du système S . L'action de S peut ne pas être instantanée mais W_β^S est la résultante de son action et de l'activité autonome de E depuis W_α^S .

Soit $W_\alpha^S \in \mathcal{R}_w$ l'état du monde du point de vue de S . Un état du monde du point de vue de S peut être décomposé en trois parties $W_{\alpha a}^S$, $W_{\alpha i}^S$, $W_{\alpha c}^S \in \mathcal{R}_w$ telles que :

$W_{\alpha a}^S$ est la partie antinomique de W_α^S

$W_{\alpha i}^S$ est la partie indifférente de W_α^S

$W_{\alpha c}^S$ est la partie coopérative de W_α^S

Nous pouvons alors dire que $W_\alpha^S = W_{\alpha a}^S \cup W_{\alpha i}^S \cup W_{\alpha c}^S$.

Soit \mathcal{R}_s l'ensemble des composants ou parties d'un système S . $\mathcal{R}_s = \{p_i \in S \text{ avec } i \in \mathbb{N}\}$.

4.2.5. Les différents états d'un système

A partir des notations précédentes, nous allons proposer les définitions formelles d'un système fonctionnellement adéquat, d'un système coopératif et d'un système à milieu intérieur coopératif.

4.2.5.1. Système fonctionnellement adéquat

Soit SFA, l'ensemble des systèmes fonctionnellement adéquats de W . Un système est fonctionnellement adéquat (cf. introduction) s'il réalise la "bonne" fonction dans son environnement.

$$\text{SFA} = \{S \in W \text{ pour lesquels } [\forall (W_{\alpha}^S, W_{\beta}^S) \in \mathcal{R}_W \times \mathcal{R}_W \mid f_s(W_{\alpha}^S) = W_{\beta}^S] \\ \text{et } W_{\alpha a}^S = \emptyset \wedge W_{\beta a}^S = \emptyset \}$$

Autrement dit, c'est l'ensemble des systèmes du monde pour lesquels tous les états du monde générés par le système ou l'environnement du système ne sont pas antinomiques pour l'autre.

4.2.5.2. Système coopératif

Soit SC l'ensemble des systèmes coopératifs. Un système est coopératif (cf. introduction) s'il est en permanence en interactions coopératives avec son environnement.

$$\text{SC} = \{S \in W \text{ pour lesquels } [\forall (W_{\alpha}^S, W_{\beta}^S) \in \mathcal{R}_W \times \mathcal{R}_W \mid f_s(W_{\alpha}^S) = W_{\beta}^S] \\ \text{et } W_{\alpha i}^S = \emptyset \wedge W_{\alpha a}^S = \emptyset \text{ et } W_{\beta i}^S = \emptyset \wedge W_{\beta a}^S = \emptyset \}$$

Autrement dit, c'est l'ensemble des systèmes du monde pour lesquels tous les états du monde générés par le système ou l'environnement du système ne sont ni antinomiques ni indifférents pour l'autre.

4.2.5.3. Système à milieu intérieur coopératif

Soit SMIC l'ensemble des systèmes à milieu intérieur coopératif de W . On suppose, un système S composé d'un ensemble de parties p_i . Chaque partie réalise une fonction f_i . La fonction globale f_S que réalise le système S dépend à la fois des fonctions partielles f_i et de l'organisation de ses parties. Chaque partie est en interaction soit avec d'autres parties, soit avec son environnement.

L'environnement d'une partie, par analogie à la définition donnée en introduction, représente tout ce qui est extérieur à cette partie, c'est-à-dire les autres parties du système et une partie de l'environnement de celui-ci. Dans la figure 4.2.5.3., l'environnement de p2 est p1, p3, p4, celui de p3 est p2, p4 et une partie de l'environnement du système...

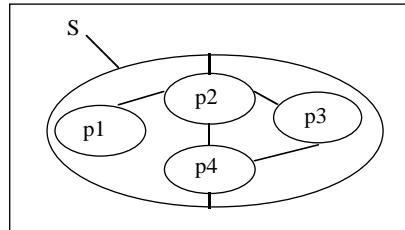


Figure 4.2.5.3. : Les parties d'un système

Une partie ne possède qu'une vue partielle de son environnement : elle ne connaît que les parties auxquelles elle est reliée par une connaissance d'acointance. Soit $W_{\alpha}^{p_i} \in \mathcal{R}_w$ un état du monde particulier du point de vue de la partie $p_i \in \mathcal{R}_s$ du système S . L'état du monde, du point de vue d'une partie, correspond par conséquent à l'état des parties qu'elle connaît et à une partie de l'environnement du système.

Le milieu intérieur d'un système est l'ensemble des parties et des supports physiques utilisés pour les échanges. Nous dirons qu'un système a un milieu intérieur coopératif si chacune de ses parties est en interaction coopérative. Une partie est en interaction coopérative si ses interactions spécifiques avec l'environnement du système sont coopératives de son point de vue.

Une partie ne perçoit et n'agit que localement sur l'environnement dans lequel elle est immergée. On définit alors :

$$\text{SMIC} = \{S \in W \text{ pour lesquels } [\forall p_i \in \mathcal{R}_s, \forall (W_{\alpha}^{p_i}, W_{\beta}^{p_i}) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_i(W_{\alpha}^{p_i}) = W_{\beta}^{p_i}] \\ \text{et } W_{\alpha_i}^{p_i} = \emptyset \wedge W_{\alpha_a}^{p_i} = \emptyset \text{ et } W_{\beta_i}^{p_i} = \emptyset \wedge W_{\beta_a}^{p_i} = \emptyset \}$$

L'état du monde correspond ici à l'état du monde du point de vue d'une partie (extérieur du système et autres parties). Dans ces conditions, un système est à milieu intérieur coopératif si chacune de ses parties est coopérative au sens où nous l'avons défini au paragraphe 4.2.5.2.

4.2.6. Approche adoptée pour la démonstration de la propriété

La démonstration se compose de trois étapes conformément à la figure 4.2.6.

L'étape ❶ consiste à établir les relations existantes en SC et SFA. Nous démontrerons tout d'abord que $SC \subset SFA$, puis que $g : SC \rightarrow SFA$ est surjective ($\forall S \in SFA, g(S')=S$ admet au moins une solution).

L'étape ❷ consiste à établir les relations existantes en SMIC et SC. Nous prouverons tout d'abord que $SMIC \subset SC$, puis que $h : SMIC \rightarrow SC$ est surjective ($\forall S \in SC, h(S')=S$ admet au moins une solution).

L'étape ❸ se déduira simplement des deux étapes précédentes.

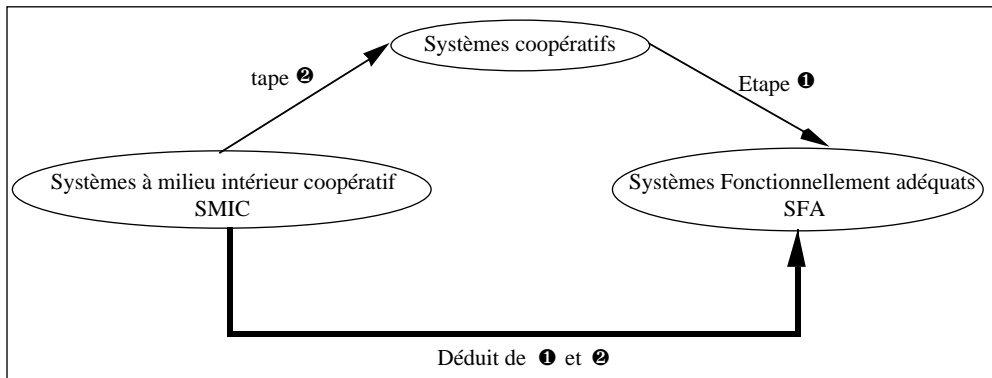


Figure 4.2.6. : Les différentes étapes de la démonstration

4.3. RELATIONS ENTRE SC ET SFA

Nous allons à présent montrer les propriétés qui existent entre l'ensemble des systèmes coopératifs et l'ensemble des systèmes fonctionnellement adéquats.

4.3.1. $SC \subset SFA$

Lemme 1 : Tout système coopératif est fonctionnellement adéquat ($\forall S \in SC, S \in SFA$)

Démonstration par l'absurde :

Hypothèse : Soit S un système tel que $S \in SC$ et $S \notin SFA$

Soit $W_\alpha^S \in \mathcal{R}_w$ un état du monde dans lequel est situé S , et $W_\beta^S \in \mathcal{R}_w \mid f_s(W_\alpha^S) = W_\beta^S$.

Conformément à la définition de SC, si $S \in SC$ alors

$\forall (W_\alpha^S, W_\beta^S) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_s(W_\alpha^S) = W_\beta^S, W_{\alpha a}^S = \emptyset \wedge W_{\alpha i}^S = \emptyset$ et $W_{\beta a}^S = \emptyset \wedge W_{\beta i}^S = \emptyset$

Si $S \notin SFA$ alors il existe au moins un couple $(W_\alpha^S, W_\beta^S) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_s(W_\alpha^S) = W_\beta^S$ pour lequel $W_{\alpha a}^S \neq \emptyset \vee W_{\beta a}^S \neq \emptyset$

On a donc ou ($W_{\beta a}^S \neq \emptyset$ et $W_{\beta a}^S = \emptyset$) ou ($W_{\alpha a}^S \neq \emptyset$ et $W_{\alpha a}^S = \emptyset$) \Rightarrow **absurde !**

Donc $\forall S \in SC, S \in SFA$ ce qui s'exprime également $SC \subset SFA$ • **CQFD**

L'ensemble des systèmes coopératifs est donc inclus dans l'ensemble des systèmes fonctionnellement adéquats (figure 4.3.1.a.).

Remarque : Malgré le résultat du lemme 1, il se peut qu'un système non coopératif soit fonctionnellement adéquat. Le système est qualifié de non coopératif car il n'est pas en interactions coopératives permanentes avec son environnement.

Le résultat du lemme 1 est donc insuffisant à double titre :

- le nombre de systèmes coopératifs peut être très insignifiant par rapport au nombre de systèmes fonctionnellement adéquats dans le monde, ce qui impliquerait qu'on ait très peu de possibilités de construire des systèmes artificiels qui soient simultanément coopératifs et fonctionnellement adéquats.

- dire d'un système coopératif qu'il est fonctionnellement adéquat est certes intéressant, mais n'est pas exploitable du moment qu'aucun algorithme de passage d'un système non coopératif à un système coopératif n'est établi.

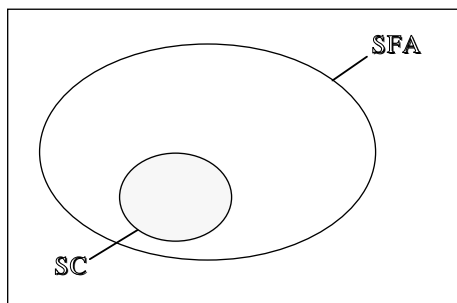


Figure 4.3.1.a. : $\forall S \in SC, S \in SFA$

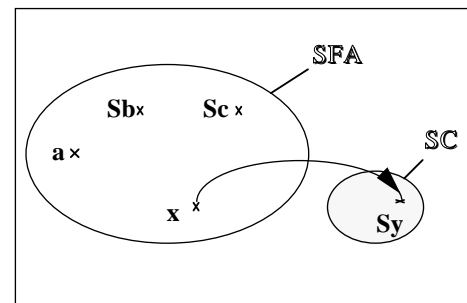


Figure 4.3.1.b. : $\forall S_x \in SFA, \exists S_y \in SC$

Conformément à notre propriété nous nous intéressons uniquement à l'ensemble des systèmes coopératifs. Pour rester dans ce cadre d'étude, l'idéal serait que pour tout système fonctionnellement adéquat il soit possible de trouver au moins un système coopératif qui soit fonctionnellement adéquat dans le même environnement. Cela signifierait qu'un système S_x (non coopératif et pourtant fonctionnellement adéquat) puisse être ramené à un système S_y coopératif réalisant une fonction équivalente (cf. figure 4.3.1.b.). Cela revient donc à montrer qu'il existe une fonction surjective f de SC dans SFA .

Ceci fait l'objet du lemme 2.

4.3.2. Surjection de SC \rightarrow SFA

Lemme 2 : Pour tout système fonctionnellement adéquat, il existe au moins un système coopératif qui soit fonctionnellement adéquat dans le même environnement.

Démonstration :

Hypothèse de départ : Soit S_x (cf. figure 4.3.1.b) un système fonctionnellement adéquat qui n'est pas coopératif c'est-à-dire $S_x \in \text{SFA}$, $S_x \notin \text{SC}$. Ce système S_x réalise une certaine fonction f_{S_x} définie par :

$$\forall (W_\alpha^{S_x}, W_\beta^{S_x}) \in \mathcal{P}_w \times \mathcal{P}_w \mid f_{S_x}(W_\alpha^{S_x}) = W_\beta^{S_x} \text{ avec } \begin{cases} W_\alpha^{S_x} = W_{\alpha i}^{S_x} \cup W_{\alpha a}^{S_x} \cup W_{\alpha c}^{S_x} \\ W_\beta^{S_x} = W_{\beta i}^{S_x} \cup W_{\beta a}^{S_x} \cup W_{\beta c}^{S_x} \end{cases}$$

Or, par hypothèse $S_x \in \text{SFA} - \text{SC}$ donc on a $W_\alpha^{S_x} = W_{\alpha i}^{S_x} \cup W_{\alpha c}^{S_x}$ et $W_\beta^{S_x} = W_{\beta i}^{S_x} \cup W_{\beta c}^{S_x}$

De plus, par hypothèse, $S_x \notin \text{SC}$, donc il existe au moins un état du monde qui n'est pas coopératif soit du point de vue du système ($W_\alpha^{S_x} \in \mathcal{P}_w$ cas ❶) soit du point de vue de l'environnement dans lequel le système est immergé ($W_\beta^{S_x} \in \mathcal{P}_w$ cas ❷) soit encore des deux points de vue (cas ❶ et ❷).

❶

• Si un état du monde est non coopératif du point de vue du système ($W_\alpha^{S_x} = W_{\alpha i}^{S_x} \cup W_{\alpha c}^{S_x}$) alors il faut modifier f_{S_x} pour que la nouvelle fonction ne prenne que $W_{\alpha c}^{S_x}$ en entrée.

Par définition, $W_{\alpha i}^{S_x}$ ($W_{\alpha i}^{S_x} \neq \emptyset$) n'est pas utile pour l'obtention de l'état $W_\beta^{S_x}$. Il est donc possible de définir f'_{S_x} une restriction de f_{S_x} telle que :

$$f'_{S_x} : W_{\alpha c}^{S_x} \rightarrow W_\beta^{S_x} \text{ avec } W_\beta^{S_x} = W_{\beta i}^{S_x} \cup W_{\beta c}^{S_x}$$

f'_{S_y} représente l'activité d'un système S_y passant d'un état $W_\alpha^{S_y} = W_{\alpha c}^{S_x}$ à un état $W_\beta^{S_y} = W_\beta^{S_x}$ et réalisant la même fonction que S_x . $f'_{S_y}(W_\alpha^{S_y}) = W_\beta^{S_y}$ avec $W_\beta^{S_y} = W_{\beta i}^{S_y} \cup W_{\beta c}^{S_y}$

Nous conservons par la suite les notations concernant le système S_x mais il ne faut pas perdre de vue que le système S_x a évolué et n'est donc plus le même qu'au début de la démonstration.

• Si dès le départ $W_{\alpha i}^{S_x} = \emptyset$, donc si l'état du monde est initialement coopératif pour S_x , le résultat de cette étape est $f'_{S_x} = f_{S_x}$

②

• Si l'état du monde est non coopératif du point de vue de l'environnement ($W_{\beta}^{S_x} = W_{\beta_i}^{S_x} \cup W_{\beta_c}^{S_x}$) alors il faut modifier la fonction f'_s pour qu'elle génère $W_{\beta_c}^{S_x}$ en sortie.

Comme $W_{\beta_i}^{S_x}$ ($W_{\beta_i}^{S_x} \neq \emptyset$) n'a aucune influence sur l'activité ultérieure de E, on peut à nouveau transformer f'_{S_x} en f'_{S_y} telle que :

$$f'_{S_y} : W_{\alpha_c}^{S_y} \rightarrow W_{\beta_c}^{S_y}$$

f'_{S_y} représente l'activité d'un système S_y passant d'un état $W_{\alpha_c}^{S_y} = W_{\alpha_c}^{S_x}$ à un état $W_{\beta_c}^{S_y} = W_{\beta_c}^{S_x}$ et réalisant la même fonction que S_x . $f'_{S_y}(W_{\alpha_c}^{S_y}) = W_{\beta_c}^{S_y}$ avec $W_{\beta_c}^{S_y} = W_{\beta_c}^{S_x}$

Pour tout système S_x fonctionnellement adéquat et non coopératif nous sommes arrivés à construire un système S_y équivalent à S_x .

• Si dès le départ $W_{\beta_i}^{S_x} = \emptyset$, donc si l'état du monde est initialement coopératif pour l'environnement de S_x , le résultat de cette étape est $f'_{S_x} = f'_{S_x}$

On réitère les étapes ① et/ou ② jusqu'à ce qu'il n'y ait plus de situations non coopératives entre le système et son environnement.

Remarques :

a - L'obtention d'un nouvel état du monde (à chaque étape ① ou ②) peut éventuellement engendrer de nouvelles activités, mais elles ne remettent pas en cause la catégorie à laquelle appartient l'ancien état, autrement dit, elles n'engendrent pas d'états antinomiques ou indifférents. Ces actions ne sont finalement effectuées que pour aboutir à un état encore plus coopératif si toutefois cela est possible.

b - D'autre part, l'ensemble des états possibles du monde est fini, même s'il peut parfois posséder un cardinal très élevé. Cette affirmation est conforme aux théories physiques courantes qui considèrent que le monde est constitué d'un très grand nombre de particules, aussi petites soient elles, qui existent en nombre fini dans l'univers. Les situations non coopératives étant un sous-ensemble des états du monde possibles, leur nombre est donc lui aussi fini.

Ces deux remarques (a et b) garantissent la terminaison de l'algorithme présenté.

$$\forall S_x \in SFA, \exists S_y \in SC \quad \bullet \text{ CQFD}$$

Le fait qu'il existe une opération de surjection de SC vers SFA signifie qu'il existe au moins un système coopératif équivalent à un système fonctionnellement adéquat dans un environnement donné. Nous allons donc nous focaliser sur ce type particulier de systèmes pour la suite de la démonstration.

Le lemme 2 répond à la première critique énoncée suite au lemme 1. Il peut exister beaucoup de systèmes fonctionnellement adéquats sans qu'ils soient coopératifs, mais le lemme 2 prouve que pour tout système fonctionnellement adéquat, il est toujours possible de le ramener à un système coopératif réalisant la même fonction.

La seconde critique n'est néanmoins toujours pas supprimée ; c'est d'ailleurs l'objet des lemmes 3 et 4.

4.4. RELATIONS ENTRE SMIC ET SC

Nous allons à présent montrer les propriétés qui existent entre l'ensemble des systèmes à milieu intérieur coopératif et l'ensemble des systèmes coopératifs.

4.4.1. SMIC \subset SC

Lemme 3 : Tout système à milieu intérieur coopératif est coopératif. ($\forall S \in \text{SMIC}, S \in \text{SC}$)

Démonstration par l'absurde :

Hypothèse : Soit S un système tel que $S \in \text{SMIC}$ et $S \notin \text{SC}$

Conformément à la définition de SMIC, si $S \in \text{SMIC}$ alors

$$\forall p_i \in \mathcal{R}_S, \forall (W_\alpha^{p_i}, W_\beta^{p_i}) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_i(W_\alpha^{p_i}) = W_\beta^{p_i}, W_{\alpha_i}^{p_i} = \emptyset \wedge W_{\alpha_a}^{p_i} = \emptyset \text{ et } W_{\beta_i}^{p_i} = \emptyset \wedge W_{\beta_a}^{p_i} = \emptyset$$

Si $S \in \text{SMIC} - \text{SC}$ alors il existe au moins une $p_i \in \mathcal{R}_S$ pour laquelle $\exists (W_\alpha^{p_i}, W_\beta^{p_i}) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_i(W_\alpha^{p_i}) = W_\beta^{p_i}, W_{\alpha_a}^{p_i} \neq \emptyset \vee W_{\beta_a}^{p_i} \neq \emptyset$

On a donc ou ($W_{\beta_a}^{p_i} \neq \emptyset$ et $W_{\beta_a}^{p_i} = \emptyset$) ou ($W_{\alpha_a}^{p_i} \neq \emptyset$ et $W_{\alpha_a}^{p_i} = \emptyset$) \Rightarrow **absurde !**

Ce lemme 3 pouvait également se déduire plus simplement.

En effet, si S est un système à milieu intérieur coopératif, cela signifie que toutes ses parties sont respectivement en interactions coopératives avec leur environnement. Or l'environnement de S , conformément à la définition qui est donnée en introduction, est l'intersection des environnements respectifs de ses parties : tout ce qui est extérieur au système.

Comme l'ensemble des interactions du système n'est rien d'autre que l'ensemble des interactions de ses parties avec l'environnement du système, qui elles sont coopératives avec leur environnement, nous pouvons immédiatement en déduire que le système a des interactions coopératives avec son environnement et donc que S est coopératif.

Donc $\forall S \in \text{SMIC}, S \in \text{SC}$ ce qui s'exprime également $\text{SMIC} \subset \text{SC}$ • **CQFD**

L'ensemble des systèmes à milieu intérieur coopératif est donc inclus dans l'ensemble des systèmes coopératifs.

Remarque : Bien que nous ayons montré que tout système à milieu intérieur coopératif soit coopératif, ce qui signifie que l'ensemble des systèmes à milieu coopératif est inclus dans l'ensemble des systèmes coopératifs (cf. figure 4.4.1.a.), il se peut très bien qu'un système à milieu intérieur non coopératif soit coopératif. Un système possède un milieu intérieur non coopératif lorsque ses parties n'entretiennent pas des interactions coopératives entre elles.

Ce résultat est également intéressant car la décomposition d'un système en parties entretenant des relations coopératives semble être un début de réponse à la seconde critique faite à propos du lemme 1. Cependant, la critique perdure tant que l'on n'a pas à disposition un algorithme de construction permettant de passer d'un système à milieu intérieur non coopératif à un système coopératif. Ce dernier point correspond justement au lemme 4.

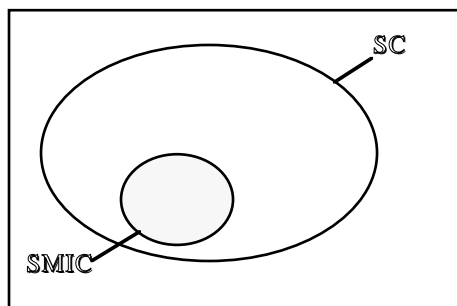


Figure 4.4.1.a. : $\forall S \in \text{SMIC}, S \in \text{SC}$

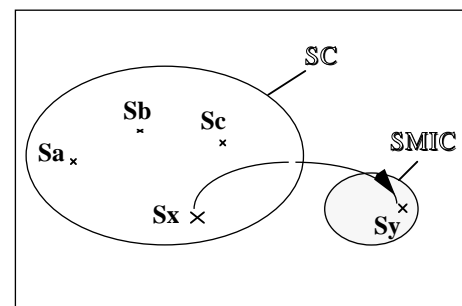


Figure 4.4.1.b. : $\forall S_x \in \text{SC}, \exists S_y \in \text{SMIC}$

Nous nous intéressons ici uniquement à l'ensemble des systèmes à milieu intérieur coopératif.

Pour rester dans ce cadre d'étude, l'idéal serait que pour tout système coopératif, il soit possible de trouver au moins un système à milieu intérieur coopératif qui soit coopératif dans le même environnement. Cela signifierait alors que le système S_x (qui n'a pas un milieu intérieur coopératif et qui est pourtant coopératif) puisse être ramené à un système S_y à milieu intérieur coopératif réalisant exactement la même fonction (cf. figure 4.4.1.b.). Cela revient à montrer qu'il existe une fonction surjective f de SMIC dans SC .

C'est l'objet du lemme 4 que nous allons démontrer ci-dessous.

4.4.2. Surjection de $\text{SMIC} \rightarrow \text{SC}$

Lemme 4 : Pour tout système coopératif, il existe au moins un système à milieu intérieur coopératif qui soit coopératif dans le même environnement.

Démonstration :

Hypothèse de départ : Soit S_x (cf. figure 4.4.1.b) un système coopératif qui ne possède pas un milieu intérieur coopératif : les parties du système n'entretiennent donc pas d'interaction coopérative entre elles.

Ce raisonnement est effectué pour chaque partie qui ne se trouve pas en situation coopérative avec son environnement.

Soit $S_x \in SC - SMIC$. $\forall p_i \in \mathcal{R}_{S_x}, \forall (W_\alpha^{p_i}, W_\beta^{p_i}) \in \mathcal{R}_w \times \mathcal{R}_w \mid f_i(W_\alpha^{p_i}) = W_\beta^{p_i}$, on a :

$$\begin{cases} W_\alpha^{p_i} = W_{\alpha_c}^{p_i} \cup W_{\alpha_i}^{p_i} \cup W_{\alpha_a}^{p_i} \\ W_\beta^{p_i} = W_{\beta_c}^{p_i} \cup W_{\beta_i}^{p_i} \cup W_{\beta_a}^{p_i} \end{cases}$$

Comme $S_x \in SC - SMIC$, on a $\forall p_i \in \mathcal{R}_{S_x}, W_\alpha^{p_i} = W_{\alpha_i}^{p_i} \cup W_{\alpha_c}^{p_i}$ et $W_\beta^{p_i} = W_{\beta_i}^{p_i} \cup W_{\beta_c}^{p_i}$

Par hypothèse $S_x \notin SMIC$; S_x ne possède pas un milieu intérieur coopératif parce qu'il existe au moins un état du monde qui n'est pas coopératif soit du point de vue de certaines parties du système S_x ($W_\alpha^{p_i} \in \mathcal{R}_w$) soit du point de vue de l'environnement de certaines parties ($W_\beta^{p_i} \in \mathcal{R}_w$) soit encore des deux points de vue.

❶

• Si un état du monde est non coopératif du point de vue d'une partie du système ($W_\alpha^{p_i} = W_{\alpha_i}^{p_i} \cup W_{\alpha_c}^{p_i}$) alors il faut modifier f_i de sorte que la nouvelle fonction ne prenne que $W_{\alpha_c}^{p_i}$ en entrée.

Par définition, $W_{\alpha_i}^{p_i}$ ($W_{\alpha_i}^{p_i} \neq \emptyset$) n'est pas utile pour l'obtention de l'état $W_\beta^{p_i}$. Il est donc possible de définir f'_i une restriction de f_i telle que :

$$\forall p_i \in \mathcal{R}_{S_x}, f'_i: W_{\alpha_c}^{p_i} \rightarrow W_\beta^{p_i} \text{ avec } W_\beta^{p_i} = W_{\beta_c}^{p_i} \cup W_{\beta_i}^{p_i}$$

f'_i représente l'activité d'une partie d'un système S_y passant d'un état $W_{\alpha_i}^{p_i}$ à un état $W_\beta^{p_i}$ et réalisant la même fonction que S_x . $\forall p_i \in \mathcal{R}_{S_y}, f'_i(W_{\alpha_c}^{p_i}) = W_\beta^{p_i}$ avec $W_\beta^{p_i} = W_{\beta_c}^{p_i} \cup W_{\beta_i}^{p_i}$

• Si dès le départ $W_{\alpha_i}^{p_i} = \emptyset$, donc si l'état du monde est initialement coopératif pour toutes les parties du système S_x , $f'_i = f_i$

❷

• Si l'état du monde est non coopératif du point du vue de l'environnement d'une partie ($W_\beta^{p_i} = W_{\beta_c}^{p_i} \cup W_{\beta_i}^{p_i}$) alors il faut modifier la fonction f'_i pour quelle génère $W_{\beta_c}^{p_i}$ en sortie.

$W_{\beta_i}^{p_i}$ ($W_{\beta_i}^{p_i} \neq \emptyset$) n'a aucune influence sur l'activité ultérieure de l'environnement de p_i , on peut à nouveau transformer f_i en f'_i telle que :

$$\forall p_i \in \mathcal{P}_{S_x}, f'_i : W_{\alpha_c}^{p_i} \rightarrow W_{\beta_c}^{p_i}$$

f'_i représente l'activité d'une partie d'un système S_y passant d'un état $W_{\alpha_c}^{p_i}$ à un état $W_{\beta_c}^{p_i}$ et réalisant la même fonction que S_x . $\forall p_i \in \mathcal{P}_{S_x}, f'_i(W_{\alpha_c}^{p_i}) = W_{\beta_c}^{p_i}$ avec $W_{\beta_c}^{p_i} = W_{\beta_i}^{p_i}$

• Si dès le départ $W_{\beta_i}^{p_i} = \emptyset$, donc si l'état du monde est initialement coopératif pour l'environnement de toutes les parties de S_x , $f'_i = f_i$

On réitère les étapes ❶ et/ou ❷ jusqu'à ce qu'il n'y ait plus de situations non coopératives entre une partie du système et son environnement.

Comme le cardinal de $\mathcal{P}_{S_y} = \mathcal{P}_{S_x}$ est fini et $W_{\beta_c}^{S_y} = \bigcap_i W_{\beta_c}^{p_i}$ on peut alors déduire que $S_y \in \text{SMIC}$.

Pour tout système S_x coopératif à milieu intérieur non coopératif nous sommes arrivés à construire un système S_y coopératif à milieu intérieur coopératif équivalent à S_x .

• Ici encore, l'obtention d'un nouvel état du monde (à chaque étape ❶ ou ❷) peut éventuellement engendrer de nouvelles activités, mais elles ne remettent pas en cause la catégorie à laquelle appartient l'ancien état : elles n'engendrent pas d'états antinomiques ou indifférents. Ces actions ne sont finalement effectuées que pour aboutir à un état encore plus coopératif si toutefois cela est possible.

• D'autre part, l'ensemble des états du monde possibles est fini (même s'il peut parfois posséder un cardinal très élevé) et il existe un nombre de parties p_i fini. Les situations non coopératives étant un sous-ensemble des états du monde possibles, leur nombre est donc lui aussi fini.

Ces deux propriétés permettent de garantir la terminaison de l'algorithme.

$$\forall S_x \in \text{SC}, \exists S_y \in \text{SMIC} \cdot \text{CQFD}$$

Le fait qu'il existe une opération de surjection de SMIC vers SC signifie qu'il existe toujours au moins un système à milieu intérieur coopératif qui soit coopératif. Ce type particulier de systèmes est celui sur lequel nous allons travailler.

Le lemme 4 répond à la dernière critique énoncée à la suite du lemme 3. En effet, nous venons de définir un algorithme permettant de passer d'un système à milieu intérieur non coopératif à un système coopératif.

4.5. RELATIONS ENTRE SMIC ET SFA

Propriété : Pour tout système fonctionnellement adéquat dans un environnement donné il existe au moins un système à milieu intérieur coopératif qui réalise une fonction équivalente.

Démonstration :

Elle se déduit immédiatement des deux lemmes précédents.

$$\left. \begin{array}{l}
 SC \subset SFA \\
 g : SC \rightarrow SFA \text{ est surjective} \\
 SMIC \subset SC \\
 h : SMIC \rightarrow SC \text{ est surjective}
 \end{array} \right\} \Rightarrow \left(\begin{array}{l}
 \exists f \mid f = g \circ h, f : SMIC \rightarrow SFA \\
 \text{avec} \\
 f \text{ surjective} \wedge SMIC \subset SFA
 \end{array} \right)$$

- a) $g : SC \rightarrow SFA$ est surjective donc $\forall S \in SFA$, il existe au moins $S_1 \in SC$ tel que $g(S_1)=S$
 - b) $h : SMIC \rightarrow SC$ est surjective donc $\forall S_1 \in SC$, il existe au moins $S_2 \in SMIC$ tel que $h(S_2)=S_1$
- De a) et b) nous pouvons déduire, $\forall S \in SFA$, il existe au moins un système $S_2 \in SMIC$ tel que $g(h(S_2))=S$. Autrement dit,

$$\exists f \mid f = goh : SMIC \rightarrow SFA \text{ surjective} \bullet \text{CQFD}$$

Ainsi, dès qu'on a un système à milieu intérieur coopératif, on a un système fonctionnellement adéquat, donc qui fournit un résultat cohérent du point de vue d'un observateur connaissant la finalité du système.

Nous avons donc :

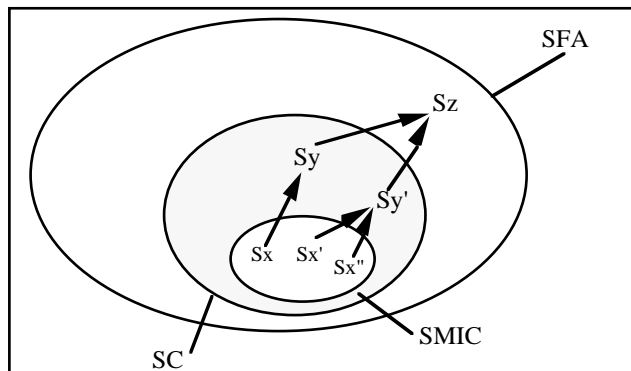


Figure 4.5. : Tout système à milieu intérieur coopératif est fonctionnellement adéquat

4.6. CONSÉQUENCES

Ce résultat est très fort dans le sens où pour tout système fonctionnellement adéquat, il est possible de trouver au moins un système à milieu intérieur coopératif qui réalise la même fonction. Ainsi, la seule préoccupation nécessaire, lorsque l'on conçoit un système, est de faire en sorte que les parties qui le constituent soient en interactions coopératives. Mais que signifie exactement coopérer ? Comment faire en sorte que les parties entretiennent des interactions coopératives ? C'est ce que nous allons étudier dans le paragraphe suivant.

En l'état actuel de la réflexion, nous savons que si toutes les parties d'un système entretiennent des relations coopératives avec leur environnement, alors le système fournit un résultat cohérent du point de vue d'un observateur capable de statuer sur l'adéquation du système.

Il nous reste à définir la notion de coopération et les mécanismes nécessaires au maintien des relations coopératives entre les parties constitutives d'un système.

4.6.1. La coopération

Bien que les démonstrations précédentes n'aient pas eu besoin de faire appel à une définition formelle de la coopération idéale, il est nécessaire d'en définir une pour pouvoir réaliser des systèmes artificiels. Ceux-ci sont en effet plongés dans un monde physique qui est le nôtre et pour lesquels nous pouvons faire des hypothèses sur ce que pourrait être une activité coopérative idéale. La coopération au niveau d'un système doit, à notre sens, au moins inclure les quatre aspects suivants :

- ① tout signal perçu par un système doit être interprété,
- ② tout signal perçu par un système doit être interprété sans ambiguïté,
- ③ l'information doit permettre au système d'aboutir à des conclusions,
- ④ enfin cette ou ces conclusion(s) doit ou doivent être utile(s) à l'environnement du système.

De manière identique, la coopération au niveau d'une partie constitutive d'un système doit, à notre sens, au moins inclure les quatre aspects suivants :

- ① tout signal perçu par une partie doit être interprété,
- ② tout signal perçu par une partie doit être interprété sans ambiguïté,
- ③ l'information doit permettre à cette partie d'aboutir à des conclusions,
- ④ enfin cette ou ces conclusion(s) doit ou doivent être utile(s) à une ou plusieurs autre(s) partie(s) du système.

Cette "définition" de la coopération signifie que chaque partie doit être bien située dans l'organisation : elle doit se trouver au bon endroit au bon moment. Une organisation vérifiant de

telles caractéristiques sera qualifiée d'**organisation optimale**. L'optimalité signifie que le système réalise bien la fonction recherchée : il est fonctionnellement adéquat.

Les parties d'un système agissant à la fois sur l'environnement et sur l'organisation du système, peuvent être assimilées à des agents (le terme agent faisant référence à la terminologie multi-agent).

4.6.2. Le mécanisme d'auto-organisation

Un système composé d'agents interagissant les uns avec les autres ne fournit un résultat cohérent que si son organisation est optimale, c'est-à-dire si les liens d'interaction entre ses agents sont conformes à la définition de la coopération idéale.

Cette optimalité des liens d'interaction entre agents peut être obtenue en dotant chaque agent d'attitudes sociales coopératives. Les attitudes sociales sont à la base de l'émergence d'un comportement collectif.

La structure que nous attribuons à un agent est composée des cinq éléments suivants :

- de **compétences**, c'est-à-dire de connaissances sur le domaine,
- de **croyances**, c'est-à-dire de connaissances qui sont souvent partiales et partielles sur les compétences d'autres agents. Il s'agit d'une "représentation que l'agent tente d'ajuster par rapport au monde" [Conte, 1995]. Les croyances décrivent l'organisation statique de la société. Une croyance exprime le point de vue que peut avoir un agent sur un autre. Les croyances relient les concepts communs à deux agents et confèrent aux agents la possibilité de raisonner sur autrui. Elles dressent les liens d'interaction possibles entre les agents de la société,
- d'**aptitudes** à raisonner à la fois sur ses connaissances et sur ses. Il s'agit plus précisément de connaissances opératoires qui peuvent consister, par exemple, à interpréter un signal en provenance d'un autre agent ou de l'environnement. L'agent doit pouvoir aussi s'adapter à la dynamique du système en apprenant sur l'organisation.
- d'un **langage d'interaction**. Les agents doivent pouvoir communiquer pour qu'émerge un comportement social à partir d'attitudes individuelles, d'où la nécessité de définir un langage d'interaction qui leur soit commun.
- d'**attitudes sociales** qui sont à la base de l'émergence d'un comportement collectif. Ces attitudes sociales sont souvent mises en oeuvre en dotant chaque agent d'un certain niveau de coopérativité. Ce dernier peut varier du degré 0 pour un agent individualiste, à un degré très important pour l'agent altruiste (il aide les autres pensant que si tout le monde a un comportement identique au sien, il en profitera lui aussi en retour).

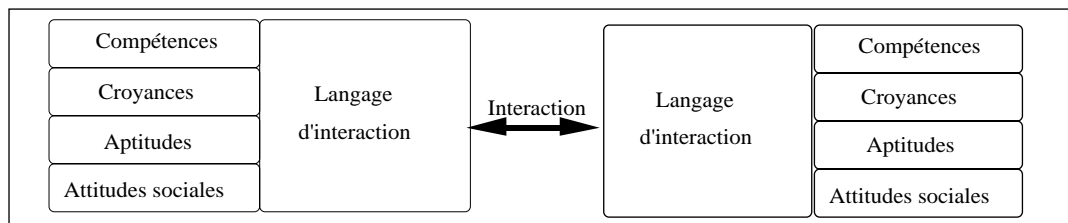


Figure 4.6.2. : Structure des agents d'ARCADIA

L'attitude sociale coopérative d'un agent conduit ici à deux activités fondamentales :

- percevoir une situation coopérative. Si tel est le cas, l'agent réalise la fonction partielle de base pour laquelle il a été conçu au sein du système.

- percevoir une situation non coopérative. Si tel est le cas, l'agent peut effectuer sa fonction partielle mais **doit** agir dans le monde de façon à supprimer cette situation et à revenir à un état coopératif. Son action a pour effet de modifier l'organisation du système en conséquence. Comme l'organisation repose sur les accointances possédées entre agent, agir de la sorte permet à l'agent de mettre à jour ses croyances.

Seul le second point demande réflexion. Il devient nécessaire de définir, en fonction du contexte, toutes les situations non coopératives qui peuvent survenir durant le fonctionnement du système ainsi que les comportements que doivent adopter les agents pour éliminer ces situations.

Il est opportun de souligner que le système ne fournit un résultat correct que si TOUS les agents qui le constituent sont coopératifs, d'où la nécessité d'encapsuler les agents dans un comportement coopératif. En faisant de la sorte, un agent n'aura pas le choix d'être coopératif ou non.

On obtient donc un système fonctionnellement adéquat en spécifiant uniquement les mécanismes de détection et de traitement des situations non coopératives, en les instanciant en fonction de l'application donnée et en encapsulant chaque agent avec le comportement coopératif défini.

4.6.3. Discussion

Notre méthode d'auto-organisation ne garantit pas une stabilité immédiate du système. Le système se réorganise en fonction des perturbations en provenance de son environnement. Il est donc amené à se réorganiser d'autant plus fréquemment que l'environnement est fortement changeant. Si l'environnement possède au contraire un nombre d'états fini, le système tendra à se stabiliser une fois tous les états vérifiés.

Le système ne transite pas d'attracteurs en attracteurs prédéfinis mais reflète au contraire la complémentarité du système avec son environnement.

Nous avons donc défini une méthode originale permettant d'obtenir des résultats cohérents du point de vue d'un observateur à partir d'un système composé de parties (agents) en interactions coopératives les unes avec les autres. Cette méthode implique de doter les agents d'attitudes sociales coopératives leur permettant, face à une situation qualifiée de non coopérative, de transformer l'organisation pour revenir à une situation coopérative.

Une situation est qualifiée de non coopérative si elle dégrade l'activité collective de la société. Elle doit être supprimée grâce à la connaissance possédée sur le collectif et par la coopération entre les agents. Plusieurs exemples peuvent illustrer la notion de coopération pour un agent :

- envoyer spontanément une information aux autres agents, s'il croit qu'elle peut leur être utile,
- accepter de satisfaire une demande explicite si elle n'est pas en contradiction avec son activité courante ou ses objectifs,
- agir sur le monde s'il croit qu'il peut favoriser les autres, même s'il n'en retire pas de bénéfice personnel.

Enfin dernier aspect non souligné explicitement jusqu'à présent mais néanmoins très important : la méthode proposée ne repose sur aucune sémantique. La définition que nous avons donnée de la coopération peut, au premier abord, faire apparaître une notion sémantique, en particulier lorsque l'on lit : "tout signal perçu par une partie doit être interprété sans ambiguïté". Pourtant, le mot interprétation signifie ici, que l'agent doit tenter de reconnaître quelque chose dans le signal qu'il vient de recevoir. Cette interprétation est purement syntaxique car locale et est grossièrement équivalente à un appariement entre le signal reçu et les compétences de l'agent. La notion d'ambiguïté fait également intervenir des notions syntaxiques et en aucun cas une quelconque notion sémantique. L'interprétation que fait un agent d'un signal qu'il vient de recevoir est effectuée de manière personnelle, selon son point de vue, sans impliquer la connaissance de ses congénères.

CHAPITRE 5

L'ALGORITHME AUTO-ORGANISATEUR

5.1. INTRODUCTION

Nous avons prouvé dans le chapitre précédent que si un système avait un milieu intérieur coopératif, il était fonctionnellement adéquat, donc fournissait un résultat cohérent. Un système possède un milieu coopératif si son organisation est optimale, c'est-à-dire, selon notre point de vue, si tout signal perçu par une partie est compris de manière non ambiguë et déclenche chez elle un certain raisonnement aboutissant à des résultats utiles à une ou plusieurs autres parties (une partie faisant référence à un ou plusieurs agents).

Entretenir de telles relations est difficile. En effet, comme les agents n'ont qu'une connaissance incomplète/erronée du monde et des autres et comme l'environnement est fortement dynamique, des situations non coopératives peuvent se produire fréquemment. Nous avons donc choisi de doter les agents d'attitudes sociales leur permettant à la fois de détecter et de traiter des situations non coopératives indépendamment de la sémantique véhiculée dans les messages échangés.

Conformément à l'hypothèse que nous avons énoncée au paragraphe 4.2.2. et sans laquelle, notre démonstration n'est plus valable, cette recherche permanente de coopération dans le but d'être fonctionnellement adéquat ne fournit des résultats corrects que si toute la connaissance nécessaire pour pouvoir obtenir un système fonctionnellement adéquat est contenue dans le système.

Pourtant, étudier l'apprentissage comme la recherche exclusive d'une organisation optimale peut sembler réducteur dans le sens où cela impose :

- que chaque agent possède un corpus de compétences fixé initialement et invariable,
- que le corpus de compétences soit complet dans le système i.e. il existe une organisation optimale qui peut fournir les comportements souhaités.

En effet, aucune organisation optimale ne peut être trouvée si le corpus de compétences des agents est incomplet. Pourtant, c'est généralement le cas dans les systèmes classiques. C'est pourquoi, nous répondons à ce problème en organisant le système multi-agent en niveaux où chaque agent est lui-même un système multi-agent composé d'agents de granularité plus faible. Ainsi, la

compétence d'un agent de niveau N change si son organisation interne (niveau N-1) est modifiée. Cette décomposition s'arrête lorsque les parties d'un système sont insécables.

Cela revient en quelque sorte à l'auto-organisation verticale dans les RANs (cf. § 3.6.1.) à ceci près que les critères de déclenchement de l'auto-organisation sont la recherche permanente de coopération avec les autres.

Cette structuration imbriquée permet de mettre en évidence les deux niveaux d'apprentissage évoqués au chapitre 2. L'apprentissage de l'organisation est réalisé par le mécanisme d'auto-organisation qui repose sur la coopération entre agents. L'apprentissage des compétences d'un agent N est effectué par auto-organisation du système multi-agent de niveau N-1 qui constitue l'agent.

Autrement dit, la méthode d'auto-organisation proposée entraîne un apprentissage des compétences grâce à une décomposition de partie en sous-parties, et un mécanisme d'apprentissage de l'organisation par une recherche permanente de coopération entre agents de même granularité.

Ce mécanisme d'apprentissage par réorganisation est fondé sur la détection et le traitement de situations non coopératives, c'est pourquoi le premier paragraphe présente six situations non coopératives génériques ainsi qu'une formalisation de celles-ci. Les deux paragraphes suivants proposent deux comportements coopératifs utiles lorsque, face à une situation non coopérative, l'agent doit agir dans le monde pour revenir à un état coopératif. Il s'agit de :

- la relaxation restreinte [Camps, 1994], [Camps, 1995], qui consiste à rediffuser de la connaissance à des voisins particuliers pour satisfaire la requête d'un agent ou fournir un résultat pertinent à un agent.
- La communication spontanée [Jennings, 1995] qui résulte d'un processus de raisonnement utilisant les croyances des autres. L'information reçue par un agent n'est pas seulement obtenue suite à une demande de sa part, mais également parce que l'émetteur pense qu'elle peut lui être utile.

5.2. LES DIFFÉRENTES SITUATIONS NON COOPÉRATIVES RECENSÉES

Ce premier paragraphe décrit six situations qualifiées de non coopératives car dégradant l'activité collective de la société. Ces situations sont présentées de manière générique et doivent être instanciées dans le domaine d'application. Il se peut d'ailleurs que certaines d'entre elles ne puissent être instanciées dans certains domaines particuliers : c'est le cas de l'incompétence dans le "tileworld" et du conflit dans le fourragement chez les fourmis. Le jeu du "tileworld" et le

fourragement chez les fourmis sont des exemples fournis en annexe qui corroborent notre méthode d'apprentissage.

Les situations non coopératives se déduisent du dual de la définition donnée au paragraphe 4.6.1., et qui considère qu'un système est coopératif si :

- ① tout signal perçu par un système peut être interprété,
- ② tout signal perçu par un système est interprété sans ambiguïté,
- ③ l'information permet au système d'aboutir à des conclusions,
- ④ enfin cette ou ces conclusion(s) est (sont) utile(s) à l'environnement.

Dans de telles conditions, un système est dit non coopératif si au moins une des quatre situations précédentes n'est pas vérifiée, auquel cas si au moins un des cas suivants est reconnu :

- a - le signal perçu par un système ne peut être interprété,
- b - le signal perçu par un système n'est pas interprété sans ambiguïté,
- c - l'information ne permet pas au système d'aboutir à des conclusions,
- d - enfin la ou les conclusion(s) déduite(s) n'est (ne sont) pas utile(s) à l'environnement.

Le cas a) équivaut à la situation non coopérative d'incompétence, le cas b) à l'ambiguïté, le cas c) à l'improductivité et le cas d) à l'inutilité. La résolution du cas d) permet de mettre en évidence deux autres situations non coopératives que nous avons appelées le conflit (cf. §5.2.7.) et la concurrence (cf. §5.2.6.).

Ces situations non coopératives peuvent se résoudre au niveau des parties constitutives du système car nous leur donnons la faculté de les détecter et d'agir dans le monde pour les supprimer. Si les parties jugent qu'elles sont en situation coopérative, cela signifie, à cet instant là, que le système est coopératif de son point de vue.

5.2.1. Notations

Nous nous plaçons dans le cadre d'un système multi-agent composé d'un ensemble fini d'agents $\mathcal{P} = \{p_1, \dots, p_n\}$

\mathcal{P}_W est l'ensemble de tous les états du monde W . $W_i \in \mathcal{P}_W (i \in \mathbb{N})$ est un état particulier du monde W . W_i^k est l'état du monde W_i observé par l'agent p_k de son point de vue.

Le prédicat "Belief $P^k[x]$ " signifie que x est une des croyances de l'agent p_k sur le monde.

L'agent p_k agit dans le monde en effectuant une fonction f_k . Du point de vue de l'agent p_k , l'action résultante de f_k dans le monde **est totalement coopérative** s'il croit qu'il y a au moins un autre agent intéressé par l'état résultat du monde et qu'aucun agent ne sera bloqué par cet état résultat. Ceci peut s'exprimer à l'aide de la formule suivante :

$$(W_{\alpha_1}^k \xrightarrow{f_k} W_{\beta_1}) \wedge \text{Belief}P^k \left[\forall p_l \in \mathcal{P} \mid (W_{\alpha_2}^l \xrightarrow{f_l} W_{\beta_2}) \wedge (W_{\alpha_2} \equiv W_{\beta_1}) \right]$$

En revanche, l'action particulière de A^k est **non coopérative** si l'agent p_k croit qu'un autre agent est incapable d'agir dans l'état résultat du monde (formule 2) :

$$(W_{\alpha 1}^k \xrightarrow{f_k} W_{\beta 1}) \wedge \text{Belief}P^k [\exists p_1 \in \mathcal{A} \mid (W_{\alpha 2}^1 \xrightarrow{f_1} W_{\beta 2}) \wedge (W_{\alpha 2} \equiv W_{\beta 1})]$$

Nous allons à présent détailler les situations non coopératives qui vont pousser chaque agent à transformer le monde pour revenir à un état coopératif.

5.2.2. La situation non coopérative d'incompétence

La situation d'incompétence correspond au cas a) du paragraphe 5.2.

- Description : L'incompétence (ou incompréhension) survient lorsque l'émetteur d'un message possède une croyance erronée sur les compétences du destinataire ou en cas d'erreur de transmission.
- Détection : Le récepteur détecte une incompétence lorsqu'il ne reconnaît rien dans le message qu'il a reçu.
- Action : Comme le message reçu ne lui est visiblement pas destiné, l'agent récepteur va (parce qu'il est coopératif donc qu'il pense que le message doit être utile à quelqu'un) tenter de le retransmettre à d'autres, qui, selon lui, pourront en tirer profit. S'il ne connaît personne susceptible d'être intéressé, il peut retourner le message à l'émetteur.
- Intérêt : Le fait de ne pas ignorer un message pour lequel on n'est pas compétent permet d'améliorer la solution collective si un destinataire pertinent est trouvé. L'effet secondaire durable est l'amélioration de la connaissance mutuelle permettant d'optimiser l'organisation, tout particulièrement lors de l'intégration d'un nouvel agent.

L'incompétence du point de vue d'un agent p_k peut être exprimée de la manière suivante :

Dans un état donné du monde W_α , l'agent p_k détecte une incompétence s'il ne peut rien extraire d'un point de vue informatif du signal perçu/reçu.
 Cette incompétence conduit l'agent p_k à essayer de faire part du signal perçu et pour lequel il ne peut rien extraire, à d'autres agents susceptibles d'être intéressés. Il ne le laisse pas de côté.

De manière générique, l'incompétence d'un agent p_k peut s'écrire :

$$\exists W_\alpha^k \mid W_\alpha^k \xrightarrow{f_k} W_\beta$$

5.2.3. La situation non coopérative d'ambiguïté

La situation d'ambiguïté correspond au cas b) du paragraphe 5.2.

- Description : Une ambiguïté survient lorsque le contenu d'un signal est incomplet, soit parce que l'émetteur a une description trop grossière des tâches du destinataire, soit parce que la spécification du message est incomplète.
- Détection : Le récepteur peut dans ce cas interpréter le message différemment et être amené à effectuer des actions potentiellement contradictoires.
- Action : L'agent pense que, dans sa société, autrui transmet spontanément des informations compréhensibles. Le message n'était pas dirigé à son intention mais il peut participer à la recherche du destinataire en le rediffusant ou bien retourner toutes les interprétations potentielles afin que l'émetteur sélectionne la plus pertinente.
- Intérêt : La suppression des ambiguïtés permet d'ajuster les croyances respectives des agents ou de découvrir des nouveaux agents plus aptes à traiter l'information.

L'ambiguïté du point de vue d'un agent p_k peut être exprimée de la manière suivante :

Dans un état donné du monde W_α , l'agent p_k détecte une ambiguïté si un même signal perçu/reçu le conduit à agir pour obtenir plusieurs états distincts du monde $W_{\beta k}, W_{\beta 1}, \dots$
 Cette ambiguïté conduit l'agent p_k à essayer de se faire aider par d'autres pour la supprimer afin d'obtenir une interprétation unique (et donc correcte du point de vue de l'agent).

De manière générique, un signal perçu/reçu est ambigu pour l'agent p_k si son interprétation l'amène à percevoir l'état du monde de manières différentes :

$$\exists W_{\alpha 1}^k, \exists W_{\alpha 2}^k \mid (W_{\alpha 1}^k \neq W_{\alpha 2}^k) \wedge (W_{\alpha 1}^k \xrightarrow{fk} W_{\beta 1}) \wedge (W_{\alpha 2}^k \xrightarrow{fk} W_{\beta 2}) \wedge (W_{\beta 1} \neq W_{\beta 2})$$

5.2.4. La situation non coopérative d'improductivité

La situation d'improductivité correspond aux cas c) et d) du paragraphe 5.2.

- Description : L'improductivité survient lorsqu'un agent possède une croyance erronée sur autrui.
- Détection : Un agent détecte une improductivité lorsqu'il reçoit un signal qu'il comprend mais dont il n'a que faire : la réception de ce signal n'entraîne chez lui aucun raisonnement ou entraîne un raisonnement n'aboutissant à aucun résultat.
- Action : Comme le message reçu ne lui est visiblement pas destiné, l'agent récepteur va (parce qu'il est coopératif donc qu'il pense que le message doit être utile à quelqu'un) tenter de le

retransmettre à d'autres, qui pourront, selon lui, en tirer profit. S'il ne connaît personne susceptible d'être intéressé, il peut le retransmettre à l'émetteur.

- Intérêt : Le fait de ne pas ignorer un message qui ne lui est pas destiné permet dans un premier temps d'améliorer la solution collective en particulier si un destinataire pertinent est trouvé. L'effet secondaire durable est l'amélioration de la connaissance mutuelle permettant d'optimiser l'organisation.

L'improductivité du point de vue de l'agent p_k peut être exprimée de la manière suivante :

Dans un état donné du monde W_α , l'agent p_k détecte une improductivité si un signal perçu/reçu est compris mais ne le conduit pas à agir dans le monde pour obtenir un état du monde $W_{\beta k}$.

Cette improductivité conduit l'agent p_k à essayer de faire part du signal perçu et pour lequel il ne peut pas agir dans le monde, à d'autres agents susceptibles d'être intéressés. Il ne le laisse pas de côté.

De manière générique, l'improductivité d'un agent p_k peut s'écrire :

$$\exists W_\beta \mid W_\alpha^k \xrightarrow{f_k} W_\beta$$

5.2.5. La situation non coopérative d'inutilité

La situation d'inutilité correspond au cas d) du paragraphe 5.2.

- Description : L'inutilité survient lorsque un agent possède des connaissances inappropriées sur autrui.
- Détection : Un agent détecte une inutilité lorsqu'il a raisonné et déduit un résultat qu'il ne peut communiquer à personne car il ne connaît pas d'agent susceptible d'être intéressé par celui-ci
- Action : L'agent peut diffuser le résultat à tous les agents qu'il connaît en espérant qu'ainsi, de proche en proche, le résultat parviendra à quelqu'un susceptible d'être intéressé.
- Intérêt : Permet de créer de nouvelles croyances entre agents partageant les mêmes intérêts.

L'inutilité du point de vue de l'agent p_k peut être exprimée de la manière suivante :

Dans un état donné du monde W_α , p_k croit qu'il n'existe aucun agent susceptible d'être intéressé par l'état résultat du monde obtenu suite au raisonnement qu'il vient d'effectuer.

Dans ce cas, p_k cherche à faire connaître son résultat par agents interposés.

L'inutilité du point de vue de l'agent p_k peut s'écrire :

$$(W_{\alpha_1}^k \xrightarrow{f_k} W_{\beta_1}) \wedge \text{Belief}P^k [\forall p_l \in \mathcal{A}, \exists W_{\beta_2} \mid (W_{\alpha_2}^l \xrightarrow{f_l} W_{\beta_2}) \wedge (W_{\alpha_2} \equiv W_{\beta_1})]$$

5.2.6. La situation non coopérative de concurrence

La situation de concurrence survient lorsque les cas ① ② ③ sont vérifiés mais pas le cas ④ . Le cas d) est alors présent. L'agent a déduit un résultat qu'un autre agent a également déduit.

- Description : La concurrence survient lorsque des agents ont des compétences similaires en raison d'une redondance initiale ou d'une duplication.
- Détection : Un agent détecte une concurrence lorsque autrui a déduit une information identique à la sienne ou lorsqu'il croit que d'autres cherchent à atteindre un but identique à l'un des siens.
- Action : Dans une société coopérative, l'activité d'un agent est en principe complémentaire des autres. Cette redondance sera profitable lorsqu'un agent n'a pu atteindre un objectif ou accepter une tâche qui lui a été demandée. Cette concurrence peut être bénéfique pour distribuer des tâches lors d'un surcroît de travail ou lors d'une incapacité ponctuelle à résoudre un problème.
- Intérêt : Le fait de distribuer certaines activités est intéressant pour répartir la charge, voire pour spécialiser des parties par l'apprentissage en évitant des duplications inutiles. La concurrence permet également d'accélérer le processus de coopération.

La concurrence du point de vue de l'agent p_k peut être exprimée de la manière suivante :

Dans un état donné du monde W_α , p_k croit qu'un autre agent p_l existe et que leurs actions non coordonnées permettent d'atteindre le même objectif (le même état du monde W_β). Ces croyances peuvent parfois être mutuelles.

Dans ce cas, p_k cherche à atteindre un nouvel état du monde afin de se coordonner avec p_l .

La concurrence du point de vue de l'agent p_k peut s'écrire :

$$(W_\alpha^k \xrightarrow{f_k} W_\beta) \wedge \text{Belief}P^k [\exists p_l \in \mathcal{A} \mid W_\alpha^l \xrightarrow{f_l} W_\beta]$$

5.2.7. La situation non coopérative de conflit

La situation de conflit survient lorsque les cas ① ② ③ sont vérifiés mais pas le cas ④ . Le cas d) est alors présent. La transformation que veut réaliser l'agent va transformer le monde dans un état qui va empêcher un (d'autres) agent(s) d'effectuer (son) leur action.

- Description : Le conflit provient généralement de l'accès multiple à des ressources communes non partageables, des informations communes incohérentes...
- Détection : Un agent détecte une contradiction si le message reçu est non ambigu mais en contradiction avec ses propres conclusions, ou s'il pense que son action dans le monde va empêcher autrui d'atteindre ses propres objectifs.
- Action : Dans une société coopérative, le conflit n'est pas volontaire, l'agent qui le détecte pense donc que l'autre n'a pas le choix. Il peut tenter de relâcher des contraintes sans contrepartie, quitte à engendrer un conflit secondaire.
- Intérêt : Chaque agent ayant ce même comportement, la négociation collective est équitable pour les agents engagés dans le conflit. Ainsi, la gestion des données partagées est cohérente, et les solutions générées satisfont au mieux les intérêts de chacun. La suppression d'un conflit permet en outre d'accélérer le processus de résolution.

Un conflit du point de vue de p_k est exprimé par :

Dans un état donné du monde W_{α} , p_k croit que son action va transformer le monde dans un état $W_{\beta k}$ tel que l'agent p_l ne pourra pas effectuer son action, parce que leurs activités respectives sont incompatibles.
 Dans cette situation, p_k doit chercher à atteindre un nouvel état du monde qui soit coopératif avec l'action que p_l doit effectuer.

Le conflit du point de vue de l'agent p_k peut s'exprimer par :

$$(W_{\alpha 1}^k \xrightarrow{f_k} W_{\beta 1}) \wedge \text{Belief}^{p_k} \left[\exists p_l \in \mathcal{A}, \exists W_{\beta 2} \mid (W_{\alpha 2}^l \xrightarrow{f_l} W_{\beta 2}) \wedge (W_{\alpha 2} \equiv W_{\beta 1}) \right]$$

Nous allons maintenant présenter deux comportements utiles aux agents pour revenir à un état coopératif.

Remarque : Certaines des situations non coopératives que nous proposons sont définies relativement aux croyances d'un agent, qui peuvent, par définition, être erronées. Le cas de plusieurs agents partageant une donnée fautive à un instant t (donc s'estiment être coopératifs entre eux) peut très bien se produire et ne met aucunement en péril la méthode d'apprentissage proposée pour les raisons suivantes :

- si l'on se place du point de vue d'un agent qui possède une croyance erronée sur un autre agent, il va agir dans le monde en fonction de sa croyance. La transformation alors effectuée dans le monde est perçue par l'agent concerné par la croyance qui détecte une situation non coopérative. L'agent initial en prendra connaissance par un feed-back ultérieur.

- si l'on se place maintenant du point de vue d'un groupe d'agents qui partagent une croyance collective erronée : le système va alors agir dans le monde et le transformer dans un état qui sera jugé non coopératif par l'environnement. Le feed-back retourné par l'environnement permettra alors à au moins un agent du système de détecter une situation non coopérative, et d'agir en conséquence.

5.3. LA RELAXATION RESTREINTE

L'auto-organisation d'une société d'agents peut être considérée comme un acte d'apprentissage collectif qui consiste en un réarrangement des éléments de cette organisation. Grâce aux interactions, les agents acquièrent de nouvelles connaissances et peuvent également connaître d'autres agents. Cette réorganisation dynamique de la société par modification des liens entre agents est un acte d'apprentissage si les modifications sont mémorisées et réutilisées.

La relaxation restreinte est une méthode permettant de modifier ces liens. Elle est basée sur la réflexion suivante : lorsqu'un agent a besoin de connaître une information particulière qu'il ne possède pas, il va rechercher des agents susceptibles de la lui fournir et dans le cas inverse, s'il possède un résultat, il va essayer de le communiquer aux agents susceptibles d'être intéressés.

5.3.1. Principe

Considérons l'exemple illustré par le schéma 5.3.1. Un agent i est bloqué en cours de raisonnement par manque d'une information particulière. Il va essayer de se la procurer en demandant de l'aide aux agents qu'il connaît et qui sont susceptibles de lui venir en aide. Il étudie pour cela ses accointances. Supposons que l'agent j soit susceptible du point de vue de l'agent i de posséder l'information manquante ; il lui envoie alors une requête signée de son nom. Supposons que l'accointance de l'agent i se révèle inexacte (la croyance n'est pas pertinente). L'agent j , parce qu'il est coopératif, ne va pas mettre la requête de côté mais, au contraire, voir s'il connaît quelqu'un de compétent. Il va donc à son tour étudier ses croyances. Supposons que les agents k et l soient jugés compétents par l'agent j . Ce dernier leur envoie donc respectivement la requête en ne changeant pas l'adresse du demandeur réel : il leur relaxe la demande. L'agent k possède réellement l'information convoitée, auquel cas, il la communique à l'émetteur du message, en l'occurrence l'agent i . L'agent i , recevant une réponse d'un agent qui, jusqu'alors, lui était inconnu, va acquérir une croyance concernant ce dernier.

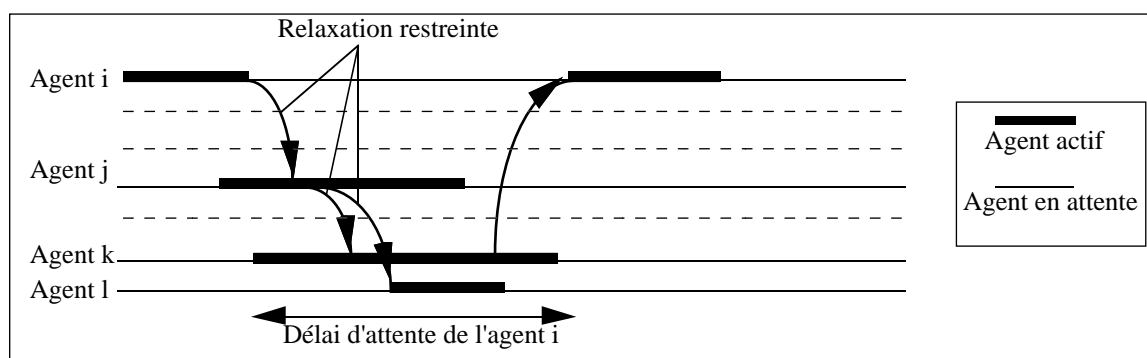


Figure 5.3.1. : Illustration de la relaxation restreinte

Ce mécanisme permet donc la création de nouvelles croyances entre protagonistes de manière autonome, ce qui correspond bien à une auto-organisation.

5.3.2. Le paradigme de la relaxation restreinte

Une méthode a été mise au point pour juger de la pertinence des agents. Elle n'exige pas de connaître la sémantique des messages mais seulement l'aptitude à distinguer des signaux élémentaires dans le signal reçu. Dans un univers de mots, par exemple, la détermination des agents pertinents est faite sur le nombre de mots communs entre ceux contenus dans les connaissances sociales de l'agent à l'initiative de la relaxation et ceux contenus dans l'information cherchée ou transmise. A chaque connaissance sociale de l'agent étant à l'initiative de la relaxation, est associé le nombre de termes qu'il a en commun avec l'information cherchée, appelé "degré de similitude". Le degré de similitude ainsi obtenu est alors comparé à un seuil fixé au préalable ; s'il lui est supérieur, l'agent est dit pertinent.

Le seuil qui permet le filtrage des agents susceptibles de satisfaire la demande ou d'être intéressés par l'information peut être calculé à partir des différents degrés de similitude obtenus de deux manières différentes :

- lorsque les degrés de similitude sont relativement proches les uns des autres, le seuil est la moyenne des degrés obtenus ;
- si en revanche, il existe un écart important entre certains degrés de similitude, on calcule la pente de la droite reliant deux degrés de similitude voisins et on la compare à une valeur que l'on se sera fixée au préalable. S'il lui est supérieur, on décide d'ignorer tous les agents ayant un degré de similitude inférieur à celui du dernier agent considéré.

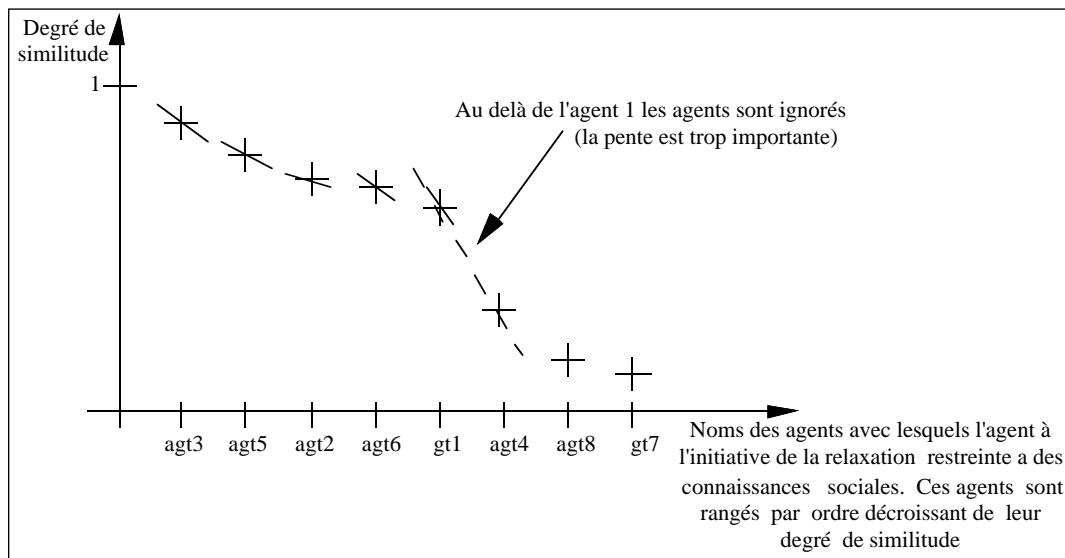


Figure 5.3.2. : Illustration du calcul du seuil permettant de déterminer les agents pertinents

Cette seconde méthode est plus restrictive que la première en particulier lorsqu'il y a plusieurs degrés de similitude de faible valeur. Une combinaison de ces deux méthodes serait peut être judicieuse.

Enfin une troisième solution, et sûrement la plus intéressante, est d'attribuer la propriété d'introspection aux agents. Chaque agent peut alors juger de la pertinence des autres agents par rapport à sa propre pertinence ce qui lui évite de raisonner sur les termes contenus dans l'information cherchée et ceux contenus dans ses connaissances sociales.

5.3.2.1. Association d'une durée de vie au message d'appel d'offre

Les messages d'appel d'offre correspondent au message émis par l'agent à l'initiative de la relaxation et aux messages propagés en cas de relaxation.

Malgré un parcours sélectif du graphe des connaissances sociales, il se peut que l'agent à l'initiative de la relaxation restreinte attende très longtemps l'information qu'il recherche ou que la diffusion de l'information qu'il a transmise soit très longue. Ceci est dû à une saturation du milieu par les messages qui transitent. Pour éviter cette saturation, une durée de vie est attribuée au message d'appel d'offre. Cette durée décroît au fur et à mesure de la recherche. Le message sera donc réémis un certain nombre de fois qui sera fixé.

Cette manière de penser est similaire aux travaux effectués dans les systèmes naturels où une fourmi qui découvre une source de nourriture informe ses congénères de sa découverte en déposant sur le sol une substance chimique, un marquage phéromonal. Cette piste va guider les autres individus de la société, qui eux-mêmes, après s'être alimentés renforceront le marquage de la piste lors du retour au nid. Tout comme les pistes chimiques que créent les fourmis sont

volatiles et disparaissent au cours du temps quand le chemin n'est plus fréquenté par les individus revenant au nid (ce qui signifie qu'il n'y a plus rien du tout), nos messages d'appel d'offre meurent au fur et à mesure de leur retransmission.

5.3.2.2. Gestion du délai d'attente local

Bien que l'on associe une durée de vie au message d'appel d'offre, l'agent à l'initiative de la relaxation restreinte peut attendre très longtemps l'information qu'il désire connaître, en particulier si l'agent recevant l'appel d'offre a un nombre important de messages à considérer avant de le traiter. Pour pallier ce problème, l'agent à l'initiative de la relaxation restreinte va limiter son temps d'attente. Si au delà du temps de recherche qu'il a autorisé, il n'a pas reçu l'information cherchée, il la considère comme inconnue. Les agents qui ne peuvent pas aider l'agent émetteur n'ont donc pas à se soucier de la manière dont il faut exprimer l'échec. En ce qui concerne la diffusion d'information, seule la durée de vie du message d'appel d'offre arrête la recherche.

L'idéal serait qu'il y ait un lien entre ces deux notions de temps, à savoir que durant la période de recherche autorisée, le message soit "relaxé" autant de fois que désiré. Pour cela l'agent à l'origine de la relaxation évalue approximativement le temps nécessaire à n redirections.

Pour permettre à l'agent demandeur de calculer localement le temps nécessaire à la recherche, le nombre moyen de messages qu'il possède à chaque instant ainsi que la durée moyenne de traitement de ses messages ont été calculés. Nous considérons que l'activité d'un agent est autonome et qu'elle s'effectue en parallèle de celle des autres.

Soit - durée_moy la durée moyenne d'un message,

- nb_msge_moy le nombre moyen de messages détenus par l'agent à l'origine de la relaxation restreinte,

- nb_réem le nombre de réémissions du message d'appel d'offre autorisé,

- nb_attente le nombre de messages moyens en attente de traitement chez l'agent demandeur avant la réponse à l'appel d'offre. ($nb_attente = nb_msge_moy$)

La formule d'évaluation du temps nécessaire par l'agent demandeur sera, dans le pire des cas, la suivante : $(durée_moy * nb_msge_moy * nb_réem) + 2(nb_attente * durée_moy)$

5.3.3. Propriétés

La relaxation restreinte ne doit pas être un comportement câblé dans chaque agent mais un comportement que chaque agent peut décider d'avoir. Elle peut ainsi être insérée dans tout système

multi-agent existant car **elle est indépendante de la sémantique véhiculée dans les messages**. Peu de modifications sont alors nécessaires si ce n'est d'associer une durée de vie aux messages d'appel d'offre donc de leur attribuer une structure de donnée différente de celle associée aux messages classiques, implanter la méthode d'évaluation locale du temps d'attente de l'agent à l'initiative de la relaxation, et enfin mettre l'adresse de l'agent à l'initiative de la relaxation restreinte dans la signature des messages d'appel d'offre.

La méthode que nous avons définie est donc une méthode d'auto-organisation générale et indépendante de l'application, dont l'implantation dans une plate-forme de développement multi-agent existante ne nécessite pas de modifications importantes.

Pour juger de sa pertinence, un simulateur a été réalisé. Il génère des sociétés d'agents avec un certain nombre d'agents et une certaine connectivité entre agents. Les connaissances sociales sont représentées par la connectivité entre agents. Chacun de ces paramètres est défini aléatoirement à chaque session. De cette manière chaque agent possède un certain nombre de connaissances sociales qui lui permet de raisonner sur autrui. Chaque agent possède également un nombre aléatoire de messages en file d'attente.

Une fois la société d'agents construite, la méthode présentée est appliquée. Les agents ne contiennent pas de connaissances explicites sur lesquelles nous pouvons nous baser pour calculer, entre autres, le degré de similitude. Ainsi lorsqu'un agent traite un message d'appel d'offre, il ne peut ni calculer le seuil permettant de définir les agents qui paraissent pertinents pour la recherche, ni définir les agents élus. Pour mettre en oeuvre ces deux étapes nécessaires, un agent génère aléatoirement un nombre entre 0 et 10 pour savoir s'il est capable d'aider l'agent demandeur ou d'être intéressé par le résultat inféré. Si ce nombre est égal à 1, la recherche est terminée. Dans le cas contraire, il doit élire une liste d'agents qui semblent pertinents pour la recherche et génère alors aléatoirement n agents parmi ses connaissances sociales. La valeur de n décroît au fur et à mesure que le message est relaxé, car nous supposons que les agents voisins sont les plus aptes à répondre à la question.

Les résultats observés sont les suivants :

- le nombre de chances de voir la recherche aboutir croît avec le nombre d'agents présents dans la société,
- il n'y a aucun risque de saturation de la société par messages d'appel d'offre, leur proportion étant minime par rapport aux messages classiques,
- le fait de limiter la durée de vie des messages d'appel d'offre n'influe pas sur la chance de voir la recherche aboutir car cette dernière a plus de chance d'aboutir lors des premières relaxations,
- enfin notre méthode locale à l'agent initiateur permettant de faire le parallèle entre les deux notions de temps que nous avons introduites donne des résultats valables dans 80 % des cas.

La relaxation restreinte s'inscrit bien dans la philosophie multi-agent car les prises de décisions des agents lors de la recherche sont locales - au niveau des agents qu'ils connaissent et qu'ils jugent pertinents - alors que le résultat fourni par cette recherche, lui, est global.

Contrairement aux autres méthodes de réorganisation dynamique ou d'auto-organisation existantes, notre méthode n'exige aucun mécanisme spécifique : elle est le fruit de la coopération entre agents. Elle ne nécessite aucun protocole particulier car il ne s'agit que d'une redirection de messages entre agents. La caractéristique essentielle est l'attribution d'une durée de vie aux messages d'appel d'offre qui n'implique pas de changements fondamentaux.

De plus, contrairement au réseau de contrats [Smith, 1981], elle est valable aussi bien pour partager des données que pour solliciter une tâche. Elle est aussi totalement asynchrone entre l'émetteur et les récepteurs potentiels ce qui évite les interblocages. L'instant du déclenchement de l'auto-organisation, ne dépend pas d'un paramètre global mais d'un manque de savoir chez un agent de la société ou d'une diffusion de résultat par un agent. Il s'agit donc bien d'auto-organisation.

Notre méthode d'auto-organisation repose sur la modification des liens entre agent par acquisition dynamique de nouvelles croyances. L'approche duale de l'apprentissage est l'oubli, qu'il serait intéressant d'exploiter lorsque les croyances sont erronées. Nous pensons que notre méthode pourrait être élargie à cet effet, notamment par la notion de délai d'attente local que nous avons présentée. Une erreur de jugement pourra alors être révisée par notre méthode de relaxation restreinte.

5.4. LA COMMUNICATION SPONTANÉE

Parce qu'un agent possède un corpus de croyances sur les compétences d'autrui, il sait si les déductions qu'il effectue peuvent intéresser d'autres agents. Ainsi, un agent coopératif communiquera de nouvelles informations aux agents qu'il estime être concernés - sans qu'aucune demande explicite ne lui soit adressée - : c'est un des aspects de l'autonomie et c'est le principe général de la communication spontanée.

Ce processus est couramment employé dans le monde naturel, par exemple lorsque des fourmis déposent des phéromones sur le chemin qui les ramène d'une source de nourriture vers la fourmilière. Ainsi, d'autres fourmis détectant ces phéromones pourront collaborer à l'approvisionnement de la collectivité. Les cibistes ont eux aussi une activité coopérative lorsqu'ils avertissent des collègues de la position de radars sur la route. On peut qualifier ce comportement d'altruiste car il ne leur est pas directement utile, mais ils espèrent un comportement ultérieur similaire des autres qui pourra leur être profitable.

5.4.1. Principe

La communication spontanée ne concerne que des agents qui partagent une donnée. Par "partager" nous sous-entendons que deux agents a et b raisonnent sur une donnée c et savent l'un et l'autre (par le biais de leurs croyances respectives) qu'ils utilisent c dans leur raisonnement. Lorsqu'un des agents modifie la valeur de cette donnée (après un certain raisonnement), il transmet la nouvelle valeur aux agents qu'il connaît et qui partagent cette donnée. La propagation de la modification se fera ainsi de proche en proche.

Un agent communique donc spontanément une donnée à un autre agent s'il croit que ce dernier partage cette donnée mais avec un état différent. Une communication spontanée peut également intervenir après une demande explicite de l'agent récepteur ; l'agent à l'origine de la communication spontanée ayant pris note de l'existence d'un agent particulier qui travaille sur une donnée commune, la lui transmettra en cas de modification.

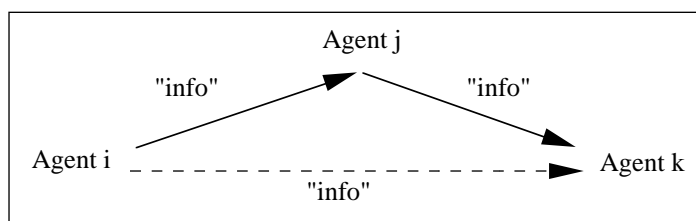
5.4.2. Mise en oeuvre

Avant d'expliquer le fonctionnement de la communication spontanée, trois définitions préliminaires s'imposent :

Définition ❶ : un agent est dans un **état stable** si et seulement si son état interne n'évolue plus sans signal d'entrée.

Définition ❷ : un **réseau des croyances est connexe** si et seulement si tout agent partage au moins une entité avec d'autres agents. Deux agents qui partagent une entité partagent aussi un chemin d'accointances sur cette entité qui les relie.

Définition ❸ : un **réseau de croyances est non redondant**, si un chemin d'accointances qui relie deux agents qui partagent une même entité est unique.



Dans un réseau de croyance non redondant, la diffusion de "info" de l'agent i vers l'agent j puis de l'agent j vers l'agent k empêche celle de l'agent i vers l'agent k.

Figure 5.4.2. : Illustration d'un réseau de croyances non redondant

Axiome : une caractéristique fondamentale de la communication spontanée est qu'un agent coopératif ne construit et ne diffuse un message que s'il est dans un état stable.

La stabilité d'un agent est relative à une action donnée. Considérons, pour illustrer nos propos, une fourmi fourrageant un espace donné. Lorsqu'elle découvre une source de nourriture, elle n'est certes pas dans un état stable puisqu'elle "prélève" aussitôt une quantité de nourriture pour la ramener à la fourmilière. Cependant, elle est dans un état stable relativement à la recherche d'une localisation de source de nourriture, et peut, à ce moment là, avertir ses congénères de sa découverte.

La communication spontanée représente un des aspects fondamentaux de l'autonomie des agents : c'est l'agent qui décide de manière autonome (sans l'intervention du développeur) de l'instant de la communication car il est le seul à savoir précisément lorsqu'il se trouve dans un état stable. Cela garantit la cohérence interne de ses données et donc que les messages qu'il va adresser à autrui auront un contenu correct (au moins temporairement du point de vue de l'agent émetteur).

Remarque : Cette propagation de la modification d'une donnée partagée risque à première vue de "boucler", notamment s'il existe des cycles. Mais ce n'est pas le cas, car seulement deux situations peuvent se présenter :

- l'agent ayant pris connaissance de la modification de la donnée ne possède pas de croyance sur autrui concernant cette donnée, auquel cas, la diffusion est stoppée,
- l'agent prenant connaissance de la modification de la donnée constate qu'il a déjà modifié celle-ci. Il a donc déjà été averti de cette modification et stoppe la propagation, car il a déjà propagé l'information lors de la première notification.

5.4.3. Propriétés

La communication spontanée permet l'obtention de deux propriétés.

Propriété n°1 : Dans une société d'agents coopératifs, si le réseau de croyances est connexe alors la gestion des données partagées est cohérente.

Si le réseau d'accointances est connexe, un agent peut supposer qu'il sera tenu au courant de toute modification d'une entité "commune", même s'il n'en a pas au préalable fait une demande explicite.

Les agents sur lesquels nous travaillons sont coopératifs (par encapsulation) et ne cherchent donc pas à induire les autres en erreur. Ainsi, chaque agent sait que toute information reçue est crédible (valable du point de vue de l'agent émetteur). La seule incertitude provient du temps qui s'est écoulé depuis l'émission. Or, même si la valeur de l'information est modifiée pendant ce Δt , l'agent récepteur sait qu'il sera tenu informé de toute modification ultérieure de l'information sans qu'il en ait fait explicitement la demande [Piquemal, 1996a] :

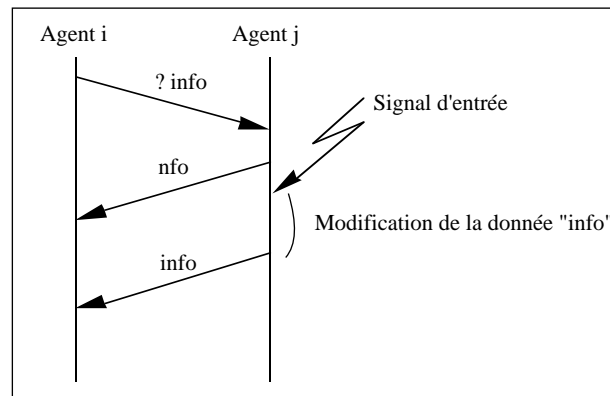


Figure 5.3.3. : La communication spontanée

Remarque : lorsque tout le SMA est dans un état stable (chaque agent est dans un état stable et il n'existe pas de communication inter-agents) alors toutes les données partagées sont cohérentes.

Propriété n°2 : Dans une société d'agents coopératifs, si le réseau de croyances est non redondant alors le nombre de messages qui transitent à chaque instant dans un SMA est minimal.

Cette propriété provient du fait que la commande d'envoi de messages n'est pas contenue dans le langage de développement d'une application, mais est un héritage implicite d'un agent. Les agents étant maîtres de leur communication, sont en mesure d'éviter les redites grâce à une analyse systématique et préalable de l'état des données à envoyer. Cela simplifie la tâche du développeur qui n'a pas à "programmer" le contenu et l'instant de l'envoi.

Tout comme la relaxation, la communication spontanée s'inscrit bien dans la philosophie multi-agent car les prises de décisions des agents lors d'un acte de communication sont locales - elles concernent les agents avec qui ils partagent une information particulière et dont le contenu vient d'être modifié - alors que les deux propriétés qui résultent, elles, sont globales.

5.5. DISCUSSION

La coopération totale que nous supposons ici n'est bien entendu possible que dans le cadre de sociétés d'agents artificiels. Mais c'est une catégorie importante, car un agent coopératif peut effectuer beaucoup d'actes, annexes à son objectif immédiat, mais utiles pour la collectivité et bénéfiques pour lui si tous les autres agents ont un comportement identique.

- La communication spontanée optimise les transferts entre agents, mais aussi simplifie le travail du développeur d'application qui n'a plus à se préoccuper de l'envoi de messages mais seulement de la gestion des croyances.

- L'agent qui reçoit un message ambigu, voire totalement incompris, ou même un message n'entraînant chez lui aucun raisonnement, va, plutôt que l'ignorer, rechercher d'autres agents plus pertinents que lui à qui le retransmettre (relaxation restreinte). Si cette recherche n'aboutit pas, il peut le retourner à l'expéditeur pour une mise à jour de ses croyances. Cela peut accroître la pertinence d'une solution globale et améliorer la connaissance mutuelle.

L'activité coopérative des agents a aussi d'autres conséquences importantes :

- La réalisation d'actes spontanés de communication garantit la gestion cohérente d'informations partagées entre plusieurs agents, lorsque le réseau des croyances est connexe.

- La théorie des actes du langage [Searle, 1979] - issue de l'étude des échanges entre les êtres humains - est très employée pour modéliser des protocoles de communications. Le fait de doter tous les agents d'une société de mêmes attitudes sociales coopératives permet d'implanter une version simplifiée de cette théorie. Les actes de la catégorie des "promissifs" ne deviennent en particulier plus indispensables.

- L'agent qui connaît un autre individu de la société avec des compétences similaires à la sienne pourra soumettre une tâche à son "concurrent" s'il est trop chargé ou s'il vient d'échouer dans celle-ci. Cela tendra à équilibrer la charge dans la société et à spécialiser les agents.

Deux aspects remarquables sont associés à l'activité auto-organisatrice que nous préconisons :

- le premier est que la théorie nous a conduits, par un simple développement logique, aux notions qui sont très fréquemment abordées (souvent isolément) dans les systèmes multi-agents, comme les conflits, la concurrence ou l'incompétence. Ce que nous apprend la théorie c'est que tous ces points doivent simultanément être pris en compte pour espérer aboutir à un système artificiel auto-organisateur.

- le second est que les activités associées à ces situations non coopératives peuvent être réellement pertinentes alors que nous n'avons fait dans leur présentation aucun pré-supposé sur le type d'application ou même la classe des problèmes abordés.

CHAPITRE 6

PRÉSENTATION DU PROJET ARCADIA

6.1. INTRODUCTION

La méthode d'apprentissage par auto-organisation présentée dans la partie précédente a été appliquée au projet ARCADIA. Ce travail s'inscrit dans le cadre d'un contrat CTI CNET n°94CND46 entre le CNET (Centre National des Télécommunications), le CERT (Centre d'Étude et de Recherche de Toulouse) et l'IRIT (Institut de Recherche en Informatique de Toulouse) [CTI, 1997], [CTI, 1996a], [CTI, 1996b], [CTI, 1995].

L'objectif du projet ARCADIA (ARchitecture pour la Coopération d'Agents D'Information Autonomes) est double :

- d'un point de vue contractuel, il consiste à élaborer une méthode, basée sur des techniques multi-agents, permettant d'accéder à de l'information pertinente dans un réseau de ressources d'information distribuées, hétérogènes et en constante évolution. Cela revient donc à permettre aux utilisateurs d'ARCADIA d'accéder à des services commerciaux électroniques de manière ludique et efficace via des opérateurs de communication comme France Télécom. Nous avons choisi de représenter chaque service et utilisateur par un agent distinct. Un agent, ainsi défini, interagit avec un ou plusieurs de ses congénères par communication directe (envoi de messages). Chaque agent ne possède qu'une vue incomplète des autres agents du fait de leur distribution et n'a qu'une connaissance partielle de leurs compétences. Ces agents sont totalement distribués dans les terminaux des utilisateurs, dans les ordinateurs des fournisseurs de services ou dans ceux des opérateurs de télécommunications.

Plus concrètement, un utilisateur abonné à ARCADIA et cherchant un renseignement particulier, émet une requête en langage naturel. Le système doit alors trouver par lui-même le ou les service(s) le(s) plus approprié(s) à la demande, si toutefois il(s) existe(nt). Il peut être éventuellement amené à faire des combinaisons de services pour répondre au mieux à la requête qui lui a été assignée. Les rôles usagers/services peuvent être symétriques en particulier si un service désire faire de la publicité ciblée.

La solution que nous soumettons doit être validée à l'aide d'un prototype. Le prototype proposé fait intervenir à la fois des services obtenus en indexant des services issus des Newsgroup, des

services issus du Web et enfin des services simulés. Tous les problèmes relatifs à l'interfaçage ont été étudiés et pris en compte par la deuxième équipe impliquée dans le projet (CERT). Bien que leur travail soit très intéressant, utile et indissociable de notre contribution [CTI, 1997], [CTI, 1996a], il ne concerne pas vraiment la problématique multi-agent de cette thèse. Nous ferons donc uniquement référence à leur contribution lors de la présentation des résultats obtenus suite à la réalisation des prototypes.

- du point de vue de cette thèse, le but du projet ARCADIA est de prouver les principes que nous avons énoncés dans la partie 2 de ce rapport. Autrement dit, nous pensons que la meilleure manière d'aborder le problème de la recherche d'information dans un contexte distribué et dynamique est de considérer le système d'information répartie comme un système auto-organisateur, dans lequel, chaque élément cherche à être en permanence coopératif avec les autres éléments du système. Plus concrètement, nous souhaitons instancier dans le cadre de ce projet la méthode d'apprentissage par auto-organisation proposée en partie 2 et qui repose sur le postulat suivant : si les entités composant un système entretiennent des relations coopératives les unes avec les autres et s'adaptent en apprenant à la dynamique inhérente au système (dynamique qui fait parfois basculer le système dans une situation qualifiée de non coopérative car dégradant l'activité collective de la société) tout en apprenant, le système fournit une solution satisfaisante au problème en cours de résolution. L'apprentissage porte essentiellement sur l'organisation : l'agent doit être situé au bon endroit et au bon moment dans l'organisation. Les compétences seront, elles aussi, modifiées grâce à la décomposition du système en sous-systèmes régis par les mêmes règles d'auto-organisation. ARCADIA sera alors composé de plusieurs niveaux.

Nous allons dans ce chapitre présenter les hypothèses du projet ARCADIA puis montrer pourquoi et en quoi les techniques multi-agents sont adaptées pour répondre à ce type de problème. Enfin, nous terminerons en présentant la méthode choisie pour permettre la recherche d'une information particulière dans un contexte distribué et dynamique ainsi que le cadre des divers prototypes qui seront réalisés.

6.2. HYPOTHÈSES DU PROJET ARCADIA

On distingue, quatre éléments primordiaux qui entrent dans la constitution d'ARCADIA et qui constituent les hypothèses de notre travail :

① - Le réseau de télécommunication : nous supposons que les utilisateurs et les services sont initialement reliés par l'intermédiaire d'un ou plusieurs réseaux afin que l'établissement d'une connexion entre les utilisateurs et les serveurs ne pose pas de problèmes techniques insurmontables.

- ② - Le système d'information répartie : la dynamique du système est très importante car il y a une perpétuelle création et suppression de services, de sites ou d'utilisateurs. D'éventuelles pannes de sites sur le réseau peuvent même survenir. En revanche, le contrôle et les informations sont distribués dans le système d'information répartie : personne ne possède de vue globale du système ainsi constitué.
- ③ - Le service : les services offerts sont très nombreux et hétérogènes. Le choix du service offert, du contenu du service, du format des informations, de la méthode d'accès aux informations relève exclusivement du fournisseur de services.
- ④ - L'utilisateur : un utilisateur ne peut pas connaître tous les services existants, ni leur mode d'emploi. Les requêtes des utilisateurs sont contenues dans des agents qui peuvent être mobiles.

ARCADIA est donc un système fortement dynamique dans lequel il est impossible d'avoir une vue globale du système tant les éléments qui le constituent sont évolutifs et nombreux. D'autre part, les services sont très nombreux et hétérogènes de par leur contenu. Chaque service exprime ses offres dans le format qu'il désire. Les usagers ne peuvent par conséquent ni connaître tous les services ni leur mode d'emploi.

Dans un tel contexte, il est impossible de déterminer l'état des paramètres du système à un instant donné : on peut donc, en ce sens, qualifier ARCADIA de "**système complexe**" [Camps, 1997a].

Avant de présenter nos premiers éléments de réponse, nous allons faire une étude comparative entre le domaine de la recherche d'information répartie et celui de la problématique multi-agent.

6.2.1. Étude comparative entre la recherche d'information répartie et les SMA

Ce paragraphe propose une étude permettant d'établir un parallèle entre la problématique des systèmes multi-agents et le domaine de la recherche d'information répartie. Nous allons pour cela étudier les caractéristiques des deux domaines de recherche dans lesquels s'insère Arcadia [Camps, 1997b]. Cinq traits fondamentaux des systèmes multi-agents se retrouvent également au niveau des systèmes d'information répartie. Ils sont énumérés dans le tableau suivant :

Système d'information répartie	Systèmes multi-agents
Les services et les usagers sont géographiquement distribués et hétérogènes.	La distribution est constitutive des SMA, tant au niveau de la connaissance qu'au niveau des traitements et du contrôle.

<p>Les usagers et les services sont très nombreux. On le voit aujourd'hui sur Internet, il y a une demande grandissante d'abonnements au système.</p> <p>Les requêtes des usagers se déplacent de manière autonome et individualiste dans le réseau pour localiser un service éventuellement pertinent. Leur migration n'est contrôlée ni par l'utilisateur, ni par le service représenté.</p> <p>Les usagers et les services ne peuvent pas connaître la totalité des services existants ni leur mode d'emploi. Ils n'ont qu'une vue locale, partielle du système auquel ils appartiennent. Ils peuvent parfois même avoir une vue erronée de leurs compétences en raison de la dynamique du système.</p> <p>Les services et les usagers interagissent pour réaliser des transactions, mais souvent de façon séquentielle et légèrement asynchrone.</p> <p>Le changement dans un tel système est très important (créations et suppressions de services et usagers). Le système pour s'adapter à cette dynamique doit être reconfiguré.</p>	<p>L'intérêt des multi-agents ne porte pas sur la manière dont un agent résout un problème donné mais sur la manière dont un groupe d'agents arrive à résoudre ce problème. L'aspect collectif est très important.</p> <p>L'autonomie des agents est un des aspects fondamentaux de la problématique multi-agent pour la résolution du problème en cours.</p> <p>La connaissance, les traitements et le contrôle sont distribués. Les agents possèdent des connaissances sur les compétences d'autrui exprimées par le biais de croyances concernant l'organisation du collectif.</p> <p>En effet, en interagissant avec les autres agents de la société, l'agent modifie l'environnement dans lequel il est immergé donc l'organisation à laquelle il appartient et ce changement entraînera une modification de la manière d'interagir pour les agents concernés par la modification de l'organisation.</p> <p>Pour faire face à la dynamique des situations les agents s'adaptent à des situations non prévues par le concepteur, ils sont capables de s'auto-organiser.</p>
---	---

Figure 6.2.1. : Similitudes entre les problématiques de la recherche d'information répartie et des systèmes multi-agents

Grâce à ce tableau, il est possible de remarquer que les contraintes inhérentes à ces deux domaines sont très proches ce qui laisse supposer que les techniques développées dans le cadre des systèmes multi-agents sont bien adaptées à la problématique de la recherche d'information. Toutefois, au-delà de l'apparente ressemblance les réponses apportées par les systèmes multi-agents reposent sur des philosophies différentes. Cette thèse s'intéresse tout particulièrement aux problèmes d'adaptation dynamique des agents. Notre objectif est de faire en sorte que dans un tel contexte, le système puisse aboutir à un

comportement collectif cohérent alors que les agents cherchent à atteindre leurs objectifs individuels alors qu'il n'existe nulle part dans le système un contrôle de leur activité fondé sur l'évaluation de la fonction globale à réaliser.

Il s'avère que pour tester les principes que nous avons avancés, le domaine de la recherche d'information dans un contexte distribué et fortement dynamique fournit un contexte idéal. Cependant, le but de cette thèse n'est pas de rivaliser avec les moteurs de recherche existants mais de mettre en évidence l'apport de la coopération dans un tel contexte : c'est la raison pour laquelle nous n'avons pas réalisé d'état de l'art sur ces derniers.

6.2.2. Les systèmes d'information coopératifs

Les systèmes d'information coopératifs sont des systèmes multi-agents appliqués à la recherche d'informations dans des bases de données physiquement distribuées et hétérogènes. L'environnement est typiquement constitué de sources d'informations variées et d'utilisateurs auxquels sont associés des agents autonomes chargés de les représenter dans le système. Un répertoire sémantique est aussi intégré dans le système pour décrire les modèles associés aux différentes sources ainsi que les relations terminologiques. L'adaptation des ressources ou des applications aux agents peut être facilitée par l'emploi de médiateurs [Wiederhold, 1992], ce qui impose l'emploi d'une ontologie pour la représentation des connaissances manipulées. Le principal problème des ontologies réside dans l'apprentissage des liens et l'acquisition de concepts. L'aspect dynamique du domaine conduit à s'intéresser fortement à l'apprentissage [Singh, 1997] :

- Classification au sein d'un agent pour l'extraction de concepts,
- Apprentissage par interaction entre les membres du groupe d'agents,
- Apprentissage par l'observation de l'activité d'autrui.

Tant par le champ d'application que l'architecture de base, Arcadia est donc clairement situé dans les systèmes d'information coopératifs. Mais il s'en distingue aussi car les limitations relevées dans ce domaine ne sont justement pas les présupposés de notre application ; à savoir celle de l'existence d'une base terminologique et ontologique préexistante. Nous considérons au contraire que l'adaptativité de ces systèmes provient de l'aptitude des agents à s'organiser globalement à partir de leurs interactions locales. La théorie que nous avons développée, devrait montrer tout son intérêt dans ce genre d'application.

6.3. APPROCHE ADOPTÉE

Nous allons mettre en évidence dans ce paragraphe la manière dont nous avons abordé le problème pour pouvoir appliquer la méthode présentée en partie 2.

6.3.1. Arcadia : un système dans un environnement

Comme nous l'avons vu au paragraphe 6.2, ARCADIA est supposé être un système en activité permanente de restructuration car composé de compétences hétérogènes et distribuées (contenues dans les services et usagers) devant être structurées.

La pression exercée par l'environnement correspond aux besoins que le système doit satisfaire. Les agents représentant les usagers et les services sont le reflet de ces besoins. Ils permettent la propagation de signaux qui vont être perçus et traités par le système. Le système doit donc tout d'abord être apte à interpréter le besoin contenu dans ce "signal". Il doit ensuite être capable de le satisfaire, puis d'agir en retour sur l'environnement en émettant à son tour un signal vers celui-ci.

A cause de la dynamique très importante, le système doit s'adapter du mieux qu'il le peut et rapidement à son environnement en faisant en sorte qu'un usager ayant fait appel à ARCADIA et satisfait du service rendu veuille de nouveau se connecter au système et que les services soient gérés de la manière la plus profitable pour les fournisseurs. Ces deux critères permettent de mesurer l'adéquation du système dans son environnement. Dans le cas contraire, une réponse insatisfaisante tendra, dans un premier temps, à encombrer le réseau par des tentatives supplémentaires de l'utilisateur. Puis peu à peu devant les échecs répétés, les utilisateurs et les serveurs feront de moins en moins appel à ARCADIA.

Outre le fait qu'il n'existe pas de bonne organisation, les changements d'organisation sont imprévisibles. Il s'agit de changements diffus de l'environnement dus aux choix des interlocuteurs entre agents communicants.

Enfin le jugement de l'adéquation est distribué et partiel. Chaque agent juge de l'adéquation requête/réponse de manière individuelle et locale.

Par voie de conséquence, ARCADIA n'est pas un système monolithique mais constitué au contraire d'un grand nombre de parties interagissant. Chacune d'elles agit sur l'environnement, sur autrui et sur l'organisation en fonction des interactions qu'elle a avec les autres : les usagers agissent sur les informations du système tandis que les réponses de celui-ci influencent les usagers. L'adéquation entre le système et les usagers sera d'autant meilleure que l'activité de médiation interne au système sera transparente aux usagers.

6.3.2. Premiers éléments de réponse

Un système d'information répartie peut être vu comme immergé dans un environnement composé d'humains qui agissent sur le système (figure 6.3.2.). Les systèmes d'information

comme Internet, Téléétel, etc., sont considérés comme externes à ARCADIA au même titre que les services et les usagers. Le processus auto-organisateur est par conséquent inopérant dans ces systèmes. Seuls les agents représentant un usager ou un service encapsulés par les comportements coopératifs que nous préconisons (cf. paragraphe 5.2) sont internes au système et s'auto-organisent en fonction des interactions.

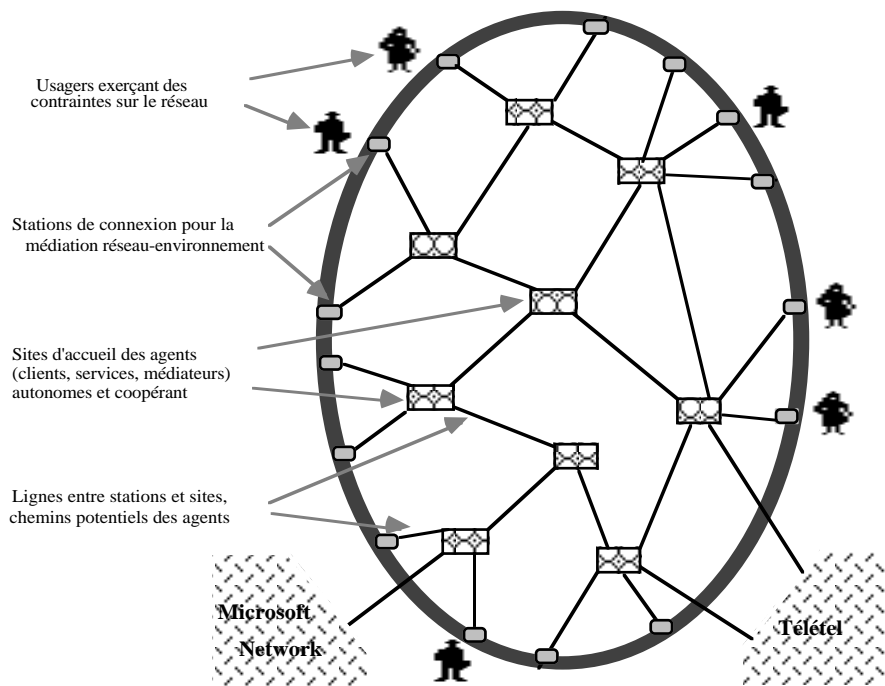


Figure 6.3.2. : ARCADIA et son environnement

La seule approche viable de ce réseau d'information est donc de le considérer **comme un tout composé de parties** (que sont les agents serveurs et usagers), **tentant de s'adapter en permanence aux contraintes de leur environnement**. Nous appliquons alors la propriété démontrée en partie 2 qui dit que pour tout système fonctionnellement adéquat il existe un système à milieu intérieur coopératif qui lui est équivalent. Un milieu intérieur est coopératif si son organisation est optimale, c'est-à-dire si tout signal perçu par une partie est compris par celle-ci de manière non ambiguë et entraîne chez elle un certain raisonnement lui permettant de déduire des informations qui doivent être utiles à une ou plusieurs autres parties. De manière idéale le système doit posséder une organisation où toute information inutile du point de vue du récepteur n'est pas communiquée, où le nombre d'agents intermédiaires pour accéder à un service particulier est minimal...

Chaque partie réalisant une fonction particulière (des besoins pour les agents usagers et des compétences pour les agents services), on conçoit que l'adaptation consiste en un processus de réorganisation permanente. Étant donnée la très forte dynamique de ce système (en nombre de

création et suppression d'agents ayant des compétences évolutives), l'adaptation doit être réalisée de manière autonome, car il serait illusoire de pouvoir contrôler à chaque instant de l'extérieur la globalité de l'organisation pour qu'elle soit optimale en permanence : chaque partie doit par conséquent se situer de manière autonome dans l'organisation. L'adaptation du système consiste alors à modifier localement la configuration préexistante compte tenu des interactions qu'effectuent les usagers et les services qui le constituent avec l'environnement. Toutes les parties du système doivent entretenir des relations telles que l'organisation à laquelle elles appartiennent, et qui est fonction de leurs interactions, soit optimale. Autrement dit, ARCADIA doit être un **système auto-organisateur** dans lequel le facteur de déclenchement de l'auto-organisation est la recherche permanente de coopération entre ses différentes parties. En s'auto-organisant, le système analyse en permanence sa structure interne pour tenter de maintenir sa "viabilité" dans l'environnement.

Le réseau sera considéré comme organisé s'il est capable d'un comportement collectif distinct de celui d'un magma constitué des mêmes éléments (agents, sites, connexions) et si les transformations pour garantir l'adaptation sont peu fréquentes et très localisées. On comprend aisément qu'un temps d'adaptation soit nécessaire et que les réorganisations soient nombreuses et fréquentes au début du fonctionnement du système. Cependant elles doivent peu à peu s'amoinrir grâce à l'apprentissage qu'effectuent les entités composant le système.

La méthode que nous proposons ne peut fournir des résultats satisfaisants que si **toutes** les parties qui composent le système sont coopératives entre elles (§ 4.5.). Si une verrue apparaissait par mégarde le fonctionnement ne serait plus du tout garanti.

Conformément à ce que nous avons montré au paragraphe 4.6.2., il faut fournir les attitudes sociales adéquates à chaque partie pour qu'elle entretienne des relations coopératives avec les autres. Avant cela, il est nécessaire d'instancier les situations dites non coopératives qui peuvent survenir durant le fonctionnement du système par rapport au domaine d'application et suivant le niveau d'organisation où l'on se situe. Ces situations sont définies au paragraphe 5.2 et leurs instanciations sont décrites dans les paragraphes 7.4.1., 8.4.1., 9.6.1.

Étant donné que notre domaine d'application fait intervenir des aspects commerciaux, rendre des agents coopératifs peut paraître très surprenant car ce n'est pas la réalité quotidienne. D'un point de vue conceptuel, l'aspect auto-organisationnel fait partie de la gestion de l'activité globale du système et n'est pas du ressort des usagers ou des services. C'est pourquoi les attitudes sociales que nous préconisons doivent être implicitement héritées au moment de la création d'un agent, plus exactement l'agent doit être encapsulé par les comportements coopératifs souhaités. Cet héritage implicite se justifie également par les deux raisons suivantes :

- dans un réseau d'information répartie, il n'y a pas de contrôle centralisé et les agents ont un important degré d'autonomie dans leurs activités ;
- le concepteur d'application (tant un utilisateur qu'un gestionnaire de service) est incapable d'associer aux agents les attitudes sociales adéquates.

6.4. ARCHITECTURE FONCTIONNELLE D'ARCADIA

Ce paragraphe expose l'architecture multi-agent que nous préconisons pour traiter le problème de recherche d'information pertinente dans un contexte distribué et fortement dynamique. Les usagers et les services qui constituent le système d'information répartie sont très dispersés géographiquement et peuvent atteindre plusieurs milliers, voire même plusieurs millions. Ils ne peuvent donc pas être regroupés sur un unique site. Afin de pallier ce problème, l'architecture proposée est composée de trois niveaux ; chaque niveau correspond à un type d'agent particulier. Voici une représentation générale de l'architecture proposée dans le cadre d'ARCADIA.

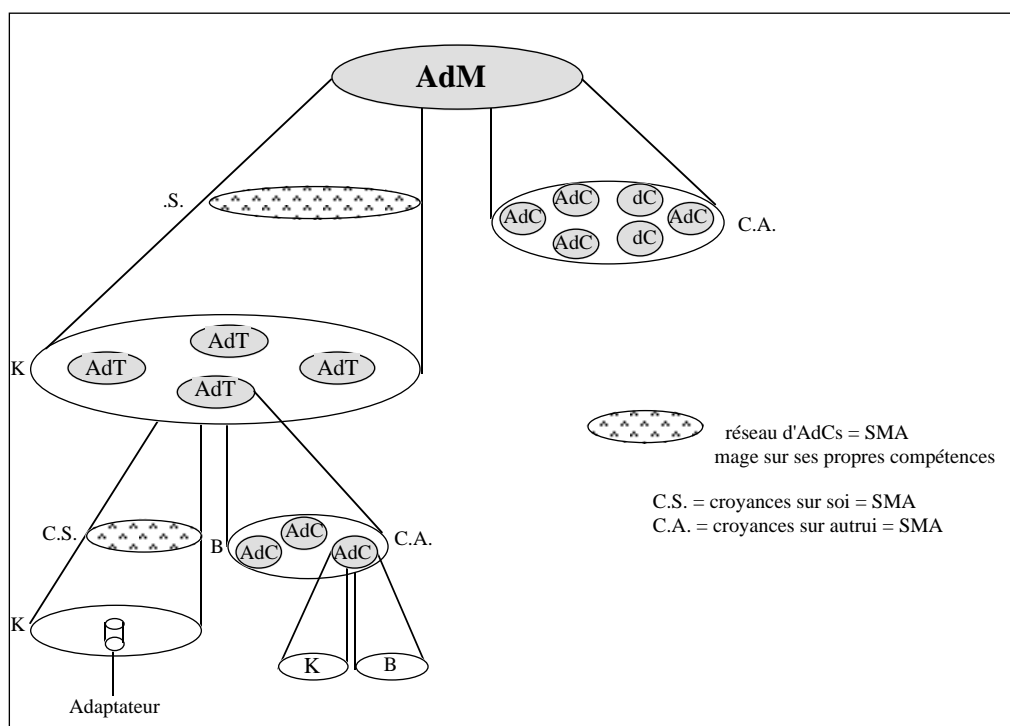


Figure 6.4. : Les trois niveaux d'ARCADIA

6.4.1. Présentation des trois types d'agents

Ce paragraphe introduit sommairement un à un chaque type d'agent défini et justifie le choix d'une architecture à trois niveaux. Chaque type d'agent est largement détaillé dans les chapitres suivants.

6.4.2. Les agents de médiations

Dans ARCADIA, chaque usager et chaque service sont représentés par un agent particulier appelé "agent représentant". Le nombre d'agents représentant étant très important, ils ne sont pas regroupés sur un site unique mais répartis sur plusieurs. Ces sites correspondent aux Agents de Médiation (AdMs). Le système d'information répartie est donc composé d'AdMs contenant chacun un ensemble d'agents représentant des usagers et des services qui interagissent les uns avec les autres. Les AdMs sont des agents sédentaires. Ils contrôlent la création et la migration des agents qu'ils contiennent. La compétence d'un AdM est l'union de tous les agents de granularité inférieure qui le composent.

Chaque AdM ne possède qu'une vue partielle des autres AdMs constituant le système. Les AdMs interagissent en fonction des connaissances qu'ils possèdent les uns sur les autres, ces connaissances leur étant propres. Il s'agit en fait de croyances qu'ils possèdent sur les compétences des autres.

ARCADIA est donc un système multi-agent constitué d'agents de médiation cherchant à s'adapter à la dynamique inhérente au système.

6.4.3. Les agents de transactions

Le niveau intermédiaire correspond aux agents de transaction (AdTs). Les AdTs peuvent être des agents représentants (AdTr) ou des agents émissaires (AdTe). Les agents représentants ont comme compétence la compétence des services ou usagers qu'ils représentent. Ils interagissent avec ces services/usagers via une interface spécifique appelée adaptateur. Les agents représentants sont sédentaires. Les agents émissaires, quant à eux, sont mobiles. Ils migrent entre AdMs selon leur besoin pour résoudre la requête qui leur a été assignée.

Un AdT possède deux états : l'état bloqué et l'état satisfait. Un AdT est bloqué lorsqu'il ne peut rien faire dans l'AdM dans lequel il est situé et lorsque la résolution de la requête qu'il a en charge n'est pas terminée. Il se trouve dans l'état satisfait lorsqu'il vient de satisfaire en totalité la requête qui lui a été assignée.

6.4.4. Les agents de croyances

Le niveau le plus bas de l'architecture proposée correspond aux agents de croyance (AdCs). Comme leur nom l'indique, ils sont chargés de représenter et de gérer les croyances possédées par les AdTs ou les AdMs. En effet, ces deux types d'agents possèdent des croyances sur leurs propres compétences et des croyances sur les compétences d'autrui. Chaque réseau de croyances est représenté par un système multi-agent constitué d'agents de croyance qui est un réseau dynamique de termes, chaque terme étant représenté par un agent particulier. De manière grossière on peut assimiler un agent de croyance à un neurone formel et un système multi-agent de croyances à un réseau neuronal.

Chaque AdT interagit avec les autres AdTs par le biais de ses croyances (connaissances sur les autres AdTs). Les AdTs situés dans chaque AdM n'ont qu'une vue partielle des autres AdTs contenus sur le même AdM.

Ainsi, l'ensemble des AdTs situés sur un même AdM constitue un système multi-agent qui tente de s'adapter à la dynamique inhérente au système.

6.4.5. Justification d'une telle architecture

Les deux premiers niveaux (AdM et AdT) de l'architecture proposée permettent de simplifier la fonction de recherche, en lançant une première exploration au niveau supérieur pour localiser le service cherché et ensuite, si la recherche aboutit, d'accéder au service proprement dit.

De plus, la migration des AdTs est facilitée car elle est mise en oeuvre par le simple mécanisme de communication entre AdMs.

Cette architecture apporte une sécurité. Les AdMs ont la faculté de contrôler les motivations des d'AdTs qui arrivent chez eux et permettent donc d'éviter d'éventuels dégâts effectués par des agents malveillants.

Le choix du troisième niveau, celui des AdCs, est justifié par notre désir de rendre le système le plus indépendant possible de l'application. Il pourrait en quelque sorte s'agir d'un vocabulaire auto-organisateur comme celui proposé par [Steels, 1996a] mais dans lequel les principes auto-organiseurs sont conformes à la propriété que nous proposons : chaque terme doit poursuivre l'unique but d'être en situation coopérative avec ses congénères.

Contrairement aux AdMs et aux AdTs, les compétences et les croyances des AdCs ne sont pas respectivement représentées par un système multi-agent. Comme nous le verrons au chapitre 9, cela vient du fait qu'un AdC représente un mot. Un AdC est par conséquent insécable. D'autre part, lorsqu'un AdC utilise ses croyances ou ses compétences, il les modifie automatiquement. L'ajout d'un niveau supplémentaire constitué d'agents de granularité inférieure n'est alors pas nécessaire pour gérer ces modifications. La décomposition en niveau est donc bornée par le bas. Un niveau de granularité supérieur à celui de l'AdM pourrait en revanche être ajouté si le nombre d'AdMs était vraiment très important. Nous nous sommes volontairement limités à ces trois niveaux car le prototype sur lequel nous testerons notre travail ne requiert pas un nombre très élevé d'AdTs. Cependant, l'ajout d'un niveau supplémentaire ne devrait pas demander de modifications insurmontables car, comme nous le verrons par la suite, le comportement des agents est générique quel que soit leur niveau de granularité.

6.4.6. ARCADIA : un système multi-agent auto-organisateur

En raison de la dynamique inhérente au système (création/suppression de sites) l'adaptation ne peut être réalisée que de manière autonome. Il serait illusoire de vouloir contrôler la globalité de l'organisation du système de sorte qu'elle soit en permanence optimale. Ainsi, le système d'information répartie est un système multi-agent auto-organisateur composé d'AdMs qui analyse en permanence sa structure interne pour s'adapter à son environnement.

Pour les mêmes raisons, un AdM est un système auto-organisateur constitué d'AdTs.

Enfin, comme l'illustre la figure 6.4.1., chaque AdM possède à la fois un réseau de croyances sur ses propres compétences (noté C.S.) et un réseau de croyances sur les compétences d'autrui (noté C.A.), tous deux étant exprimés par un système multi-agent auto-organisateur constitué d'AdCs. Il en est de même pour chaque AdT.

ARCADIA est par conséquent composé de trois types de systèmes multi-agents distincts qui s'auto-organisent en suivant exactement les mêmes principes : la recherche de coopération avec leur environnement et avec leurs congénères (qui constituent le milieu intérieur du système).

De manière générique, pour un système constitué de N sites comprenant en moyenne M AdTs, ARCADIA est constitué de $3N+2(M*N)+1$ systèmes auto-organisateurs.

6.5. CADRE DES DIVERSES DÉMONSTRATIONS

Afin de pouvoir tester notre méthode, nous avons imaginé un service à valeur ajoutée qui tente de fournir des informations de la même qualité qu'un forum de discussion et qui en corrige certains

défauts (requête d'un nouvel abonné qui a été posée peu de temps auparavant, utilisation d'archives interrogeables de manière trop générique, ...).

Le service que nous proposons est composé d'un forum de discussions traitant d'un thème particulier et de quelques services de base traitant de ce thème, accessibles sur Internet. Le but de ce service est d'offrir la possibilité à ses abonnés d'obtenir des informations complémentaires récoltées dans d'autres sources d'informations que le forum. Il doit également permettre la mise en relation des utilisateurs ayant des besoins similaires ou complémentaires. Les utilisateurs abonnés à ce service ont alors la possibilité de lire les articles pertinents qui ont déjà été postés et d'être prévenus lorsque de nouveaux articles intéressants sont postés.

Le service est implanté par une instance d'ARCADIA, c'est-à-dire une spécialisation des agents génériques d'ARCADIA pour traiter des petites annonces de véhicules. Cette spécialisation se traduit par la définition d'agents de transaction représentant des sources d'information existantes sur Internet et l'instanciation des croyances des agents par la description des thèmes des informations qu'eux-mêmes ou d'autres proposent ou recherchent.

Les divers prototypes ont été réalisés sur plate-forme SUN SPARK, fonctionnant sous UNIX System V et en utilisant l'environnement de programmation OPENWIN. Le langage de programmation utilisé (pour la partie multi-agent tout au moins) est C++. Nous avons également utilisé des bibliothèques d'objets C++ fournies par CPPV (C++ Views). Elles proposent une centaine de classes d'objets C++ dont notamment une classe implantant des listes de pointeurs sur objet (VOrdCollect), une classe permettant la gestion de chaînes de caractères (VString)... La totalité du code correspondant à l'implémentation du système multi-agent équivaut à environ 10000 lignes de codes et à une quarantaine de classes.

Les AdTs et les AdMs correspondent à des processus UNIX qui évoluent en parallèle de manière asynchrone. Les AdCs sont quant à eux des instances de classe.

6.5.1. Les différents mécanismes de communication utilisés

Ce paragraphe présente les différents mécanismes de communication permettant aux agents d'échanger de l'information.

6.5.1.1 Structure des messages échangés entre AdMs et/ou AdTs

Afin de pouvoir détailler le processus d'interprétation mis en oeuvre par tous les agents d'ARCADIA (indépendamment de leur granularité) lors de la réception d'un message, il est nécessaire de détailler la structure d'un message.

Les messages échangés entre les agents appartenant aux deux niveaux supérieurs de l'architecture proposée (AdM, AdT) sont composés de sept champs, c'est-à-dire (cf. figure 6.5.):

- d'un **expéditeur** représenté par une structure appelée *signature* qui contient le nom de l'AdM dans lequel est situé l'expéditeur, le nom de l'AdT expéditeur proprement dit et le numéro du message qu'il envoie. Cette structure permet de garantir l'unicité de la signature,
- d'un **destinataire** représenté également par une structure signature,
- d'une **force illocutoire** ('?' dans le cas d'une interrogation, '!' lors d'une affirmation, '~' lors d'une négation et '#' lors d'une ignorance). Les quatre actes de langage cités forment un sous-ensemble très restreint des actes énoncés par Searle [Searle, 1979], et sont indispensables pour percevoir l'intention associée au contenu propositionnel,
- d'un **nombre de relaxations** c'est-à-dire d'un nombre exprimant la durée de vie du message qui n'est autre que le nombre de réémissions possibles de ce message sans modifier aucun champ de sa structure (cf. 5.3.2.1.),
- du **corps du message** proprement dit,
- du **nom d'un AdT pertinent**. Cette identité est fournie par l'AdM en réponse à une sollicitation d'un de ses AdTs bloqué.
- d'un **destinataire logique**. Ce champ est utilisé lors d'une communication directe entre AdTs. Par exemple, si un utilisateur U1 veut envoyer un message personnel à un utilisateur U2, le message transite directement de l'AdT représentant U1 vers l'AdT représentant U2.

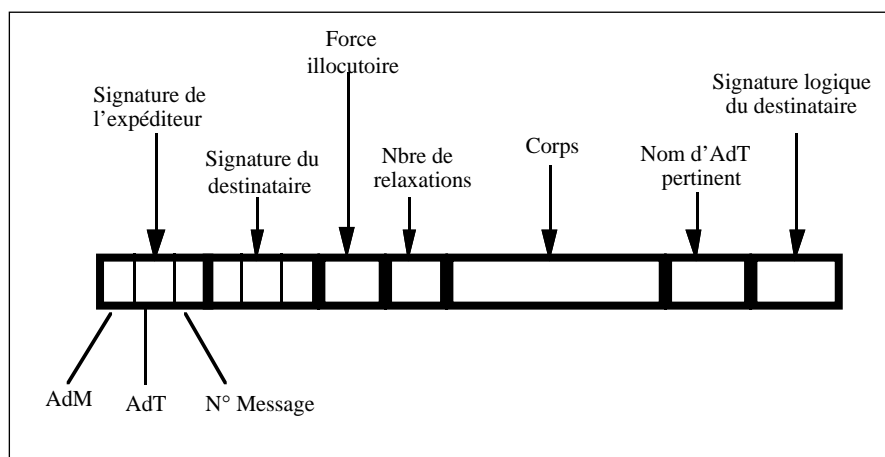


Figure 6.5.1.1. : Structure des messages dans ARCADIA

Les agents communiquent par le biais de leurs croyances ; ils ne savent pas si le destinataire courant du message est le service réellement cherché. C'est la raison pour laquelle nous avons choisi de laisser le champ destinataire vide dans le cas d'une demande de localisation d'un service précis. Ce cas est utilisé pour mettre en oeuvre la relaxation restreinte (cf. §5.3.) et la communication spontanée (cf. §5.4.).

En revanche, lorsque ce champ est rempli, il s'agit d'une communication ciblée. Si par cas le récepteur explicite du message est un AdT qui se juge incompetent ou inintéressé par le message reçu, il doit tout de même le transmettre à l'utilisateur ou au service qu'il représente. Même s'il connaît un agent complémentaire, l'agent ne relaxe pas le message par souci de confidentialité.

D'autre part, chaque agent gère une table d'objectifs contenant ceux qu'il essaie de résoudre. Lorsqu'un agent reçoit une requête (message avec '?' comme force illocutoire), s'il est compétent, il se crée un objectif correspondant à la résolution de celle-ci, le mémorise dans sa table des objectifs et effectue une action visant à la résoudre. Toute action qu'il effectue afin d'atteindre un objectif est mémorisée dans une liste d'actions associées à cet objectif. Une action consiste en fait en un acte de communication.

Un agent n'attend pas indéfiniment une réponse à une requête. Lors de la réception d'un "timeout" associé à une action particulière (lui indiquant qu'il a suffisamment attendu une réponse à l'action courante), l'agent met à jour le ou les réseau(x) de croyances concerné(s) puis supprime l'action concernée par le timeout de la liste des actions associées à l'objectif courant. S'il ne reste alors plus aucune action associée à l'objectif courant, l'objectif est également ôté de la table des objectifs : l'objectif a été atteint ou au contraire il a échoué.

D'un point de vue plus pratique un objectif comporte quatre champs :

- l'objectif lui-même qui n'est autre que la requête à résoudre,
- la(es) action(s) que l'agent a effectuée(s) pour atteindre cet objectif (qui correspondent finalement à des actes de communication). Une action est composée de 4 champs qui sont :
 - la signature de l'émetteur (afin de mémoriser le numéro du message envoyé par l'agent courant),
 - la signature du destinataire,
 - un marqueur (booléen) qui indique si le destinataire a répondu ou pas à la sollicitation,
 - un feed-back qui contient la signature du l'agent à l'origine de la réponse. Ceci est en particulier utilisé lorsqu'un AdT transmet une réponse à son adaptateur. L'AdT mémorise alors dans le champ courant la signature de l'agent qui sera concerné par le feed-back envoyé par son adaptateur,
- un booléen (Reussi) indiquant si l'objectif a été atteint ou pas,
- un booléen (solAdm) indiquant dans le cas d'un AdT si ce dernier a déjà sollicité l'aide de son AdM pour essayer d'atteindre l'objectif courant.

Pour illustrer nos propos voici l'exemple d'un objectif possédé par l'agent AdT1

Objectif de AdT1	Actions
adap 1 § AdM1 AdT1 0 § ? § req1 § § §	AdM0 AdT1 1, AdM0 0, M = !, FB =
	AdM0 AdT1 2, AdM0 AdT3 0, M = !, FB =
	AdM0 AdT1 3, AdM0 AdT2 0, M = ~, FB =
	AdM0 AdT1 4, AdM0 AdT5 0, M = ~, FB =
	AdM0 AdT1 5, adap 0, M = ~, FB = AdM0 AdT3 2
Objectif atteint : NON	Sollicitation AdM : OUI

Suite à la requête transmise par son adaptateur, AdT1 s'est créé un objectif, celui de la résoudre. Les actions qu'il a effectuées pour aller dans ce sens sont au nombre de cinq. Il a tout d'abord sollicité l'aide de son AdM qui lui a répondu. Il a ensuite envoyé la requête à trois AdTs : AdT3, AdT2 et AdT5. AdT3 lui a répondu. Il a alors transmis la réponse de ce dernier à son adaptateur et n'a pas reçu encore le feed-back de ce dernier. Le feed-back qu'il recevra concernera donc l'agent AdT3. L'objectif n'est jusqu'alors pas atteint mais AdT1 a déjà sollicité l'aide de son AdM.

6.5.1.2. Structure des messages échangés entre AdCs

La structure des messages échangés entre AdCs est différente de celle des messages échangés entre AdTs et ou AdMs.

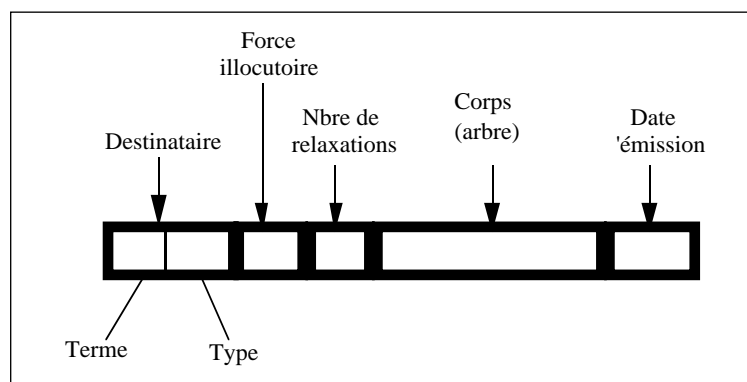


Figure 6.5.1.2. : Structure des messages échangés entre AdCs

Elle est composée de cinq champs (cf. figure 6.5.1.2.):

- d'un **destinataire** représenté par une structure appelée *Terme* qui contient le mot que représente l'AdC destinataire ainsi que le type de l'agent récepteur à savoir agent atomique, requête, service ou frontière (cf. § 9.2.),
- d'une **force illocutoire** ('?' dans le cas d'une interrogation, '!' lors d'une affirmation, '~' lors d'une négation et '#' lors d'une ignorance),
- d'un **nombre de relaxations** c'est-à-dire d'un nombre exprimant la durée de vie du message qui n'est autre que le nombre de rediffusions possibles de ce message sans modifier aucun champ de sa structure (cf. § 5.3.2.1.),
- du **corps du message** proprement dit qui est représenté par une structure appelée *Arbre* qui correspond à l'information reçue des fils sous forme structurée (cf. § 9.5.5),
- de la **date d'émission** du message afin de pouvoir simuler le parallélisme des communications.

6.5.1.3. Mécanismes d'implantation des communications inter-agents

Les AdMs représentent respectivement les différents sites du système ARCADIA sur lesquels sont regroupés les AdTs. Ces sites peuvent être situés sur une même machine mais il semble plus naturel de les considérer situés sur des machines distinctes. Les AdTs, quant à eux, sont groupés sur une seule machine.

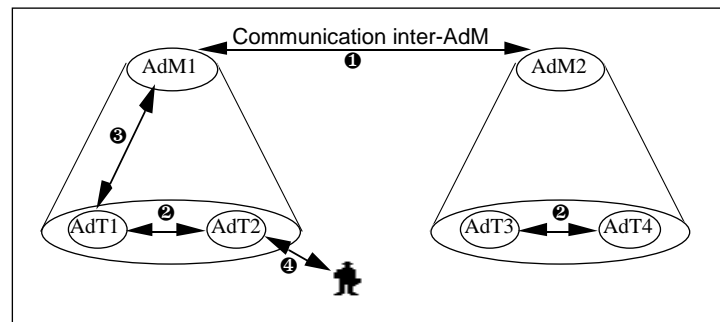


Figure 6.5.1.3. : Les différentes communications entre AdMs et ou AdTs

Les différents mécanismes de communication nécessaires à ces agents doivent donc tenir compte de ces contraintes. Conformément à la figure 6.5.1.3., les échanges inter-AdMs (cas ❶) doivent permettre des communications inter-machines, ceux inter-AdTs (cas ❷) ainsi que ceux entre AdM et AdT (cas ❸) sont locaux à une seule machine.

Plus précisément, il est possible de distinguer deux types de communication au niveau des AdTs : la communication inter-agents (communication inter-AdTs ou AdT/AdM) interne au système et la communication entre le système et son environnement (d'un AdT vers l'utilisateur ou le service qu'il représente et vice-versa - cas ❹).

Cette architecture met donc en évidence une topologie du système pour laquelle nous avons, en simplifiant, deux domaines : l'un interne (A) (cas ❶, ❷, ❸), l'autre externe (B) (cas ❹). Ceci impose un modèle de communication dans lequel nous pouvons imaginer tous les liens possibles : le cas 1 : A1 --> A2, le cas 2 : A1 --> B1 et le cas 3 : B1 --> A1. La dernière possibilité (B1 --> B2) ne concerne pas ARCADIA.

Dans le cadre de ce travail de thèse, seul le cas 1 nous incombe. D'un point de vue contractuel, les cas 2 et 3 ont du être considérés et ils ont été étudiés par l'équipe du CERT également impliquée dans ce projet.

Selon la classification de Lamport [Lamport, 1990], trois modes de communication possibles peuvent être distingués : la communication en mode message (message passing), la communication par mémoire partagée (shared memory) et la communication synchrone (synchronous

communication - RPC). Nos processus évoluant de manière asynchrone, ce dernier mode de communication n'est pas adapté à notre étude.

Pour des raisons d'architecture du système, les AdTs représentants (donc qui ne migrent pas), n'ont pas besoin de posséder un mécanisme de communication applicable aux systèmes répartis. En revanche, il n'en est pas de même pour les AdMs. En effet, ARCADIA met en jeu des processus (AdMs) pouvant s'exécuter sur des machines différentes et ne pouvant donc pas communiquer via une mémoire partagée. Cette seconde caractéristique nous a conduits à nous intéresser aux sockets. Les sockets permettent de faire communiquer aussi bien des processus s'exécutant sur un même site que des processus s'exécutant sur des sites distincts. D'autre part, le passage du modèle centralisé au modèle réparti se fait facilement moyennant quelques modifications mineures. Nous avons donc choisi de faire communiquer les AdMs (cas ❶) par le biais de sockets UNIX ou INTERNET selon qu'ils sont localisés ou non sur différentes machines. D'autre part, afin d'avoir un mécanisme de communication générique aux deux niveaux supérieurs de l'architecture, nous avons choisi de faire également communiquer les AdTs entre eux (cas ❷) et avec leur AdM (cas ❸) par le biais de sockets UNIX.

La recherche d'un service particulier (donc d'un AdT) peut se faire à l'aveugle dans le sens où l'agent ne sait ni si le service recherché existe ni par conséquent sa localisation. Cependant les adresses physiques des divers agents permettant le cheminement jusqu'à lui sont connues (soit suite aux interactions passées, soit grâce à l'aide fournie par l'AdM). Il en est de même pour la communication inter-AdM.

Les AdCs, quant à eux, communiquent par échange de messages. Ces derniers sont réalisés par l'intermédiaire d'un agenda où tous les agents déposent leurs messages dans l'ordre d'arrivée. L'agenda utilise une liste contenant un pointeur sur chaque agent afin de leur transmettre les messages par ordre croissant de date.

6.5.2. Les différentes sources d'information utilisées

Dans ce scénario, nous proposons d'utiliser ARCADIA pour améliorer les informations trouvées dans un forum tel que [fr.petites-annonces.vehicules](http://fr.petites-annonces.vehicules.com) qui permet de diffuser des petites-annonces pour l'achat ou la vente de véhicules d'occasion. Nous allons à présent détailler les services de base qui sont susceptibles d'être utilisés dans le cadre de la démonstration. Les services de base que nous considérons sont au nombre de huit. Ils proviennent des principales sources d'information en français accessibles sur le Web et les News. Les services issus du Web se nomment "Havas on line" (<http://www.hol.fr/AnnoncesClassees/AnnoncesResult.htm?typeOffres>

&rubrique=AUTOVELO ou <http://www.hol.fr/AnnoncesClassees/AnnoncesResult.htm?typeRecherche&rubrique=AUTOVELO>), “France classée” (<http://www.generation.net/~cybsup/fc/auto.htm>), “Furanet” (<http://www.impac.be/voiture.htm>), “Webdata” (<http://www.web-data.fr/annonce/consult.html>) et “Toutela” (<http://www.toutela.com/consulter/consultation.htm>). Celui issu des News s’appelle “pavehicules” (fr.petites-annonces.vehicules).

Enfin le dernier service, nommé “Argus” est une simulation d’un service de type ARGUS fabriqué de toutes pièces par le CERT. Cette source d’information analyse les petites-annonces proposées dans les différents sites pour établir le prix des véhicules en fonction de leur modèle, kilométrage et année.

Chaque service présenté est représenté par un AdT. Les réseaux de croyances possédés par les AdTs ou les AdMs entrant dans la constitution des divers prototypes ont été élaborés à l’aide des diverses annonces contenues dans les services que nous venons de présenter. A ces services présents sur le Web se rajoutent deux usagers, l’un cherchant à acheter une voiture particulière, l’autre désirant en vendre une. Le but des scénarios présentés par la suite est de mettre ces personnes en contact tout en faisant ressortir les comportements coopératifs des agents qui participent au travail.

Un adaptateur est associé à chacun des services et usagers présents dans le prototype et a fait l’objet d’étude d’une équipe du CERT (Pierre Bieber et Patrice Cros) également impliquée dans le projet ARCADIA. Le rôle d’un adaptateur est d’assurer les communications entre l’agent représentant et le service correspondant. Pour cela, chaque adaptateur traduit les requêtes provenant de l’agent représentant pour qu’elles soient compréhensibles par le service. Inversement, l’adaptateur met au format ARCADIA les réponses fournies par les services. Un adaptateur a également été développé pour chaque utilisateur. Il permet à l’utilisateur de poster des petites-annonces via un “browser” Web tel que Netscape ou Mosaïc. L’adaptateur de l’utilisateur permet également de visualiser les réponses fournies par les différentes sources d’information.

Les adaptateurs des AdTs étant localisés au CERT, un mini protocole de communication entre les deux sites IRIT et CERT a du être réalisé. Ce protocole sert entre autres aux communications d’ARCADIA vers l’extérieur et a été implémenté par le CERT.

6.6. DISCUSSION

Dans ce chapitre nous avons fait le parallèle entre la théorie décrite en partie 2 et les hypothèses du projet sur lequel nous devons l’appliquer. Nous avons tout d’abord montré que les principes multi-agents semblent être adaptés à la recherche d’information pertinente dans un contexte réparti en raison des similitudes qui existent entre ces deux domaines de recherche.

Nous avons alors ensuite étudié quelles étaient les hypothèses nécessaires pour pouvoir appliquer la propriété énoncée et démontrée en partie 2. Il nous faut tout d'abord considérer le système d'information comme composé de parties (agents ou groupes d'agents) interagissant les unes avec les autres. Conformément à la propriété 4.5, pour que le système soit fonctionnellement adéquat, il faut qu'il possède un milieu intérieur coopératif, autrement dit avoir toute ses parties coopératives entre elles.

Nous avons présenté l'architecture que nous avons définie dans le cadre du projet ARCADIA. Cette architecture comporte trois niveaux distincts, chaque niveau correspondant à un type d'agent spécifique.

Enfin, nous avons exposé les différents fondements des prototypes que nous avons réalisés.

Ces précisions étant données, nous allons pouvoir à présent étudier en détail chaque système multi-agent présent dans ARCADIA. Bien que le système multi-agent d'agents de croyance soit utilisé par les deux niveaux supérieurs de l'architecture, nous allons tout de même commencer par présenter les AdMs et les AdTs. Lorsque nous ferons référence aux réseaux de croyances (en particulier lors de l'explication du mécanisme d'interprétation chez l'AdM et l'AdT) nous spécifierons uniquement les entrées nécessaires et les sorties fournies par le système multi-agent d'agents de croyance.

CHAPITRE 7

LE SYSTÈME MULTI-AGENT D'AGENTS DE MÉDIATION

7.1. INTRODUCTION

Les agents de médiations (AdMs) correspondent aux sites réels du système d'information répartie. Un AdM est sédentaire et a pour rôle de contrôler les créations, suppressions et migrations des AdTs qu'il contient. La création d'un AdM se produit lors de l'installation d'un nouveau site ARCADIA par France Télécom. Il est alors doté des aptitudes communes à tous les agents (interprétation, communication, attitudes sociales, ...)

7.2. STRUCTURE D'UN ADM

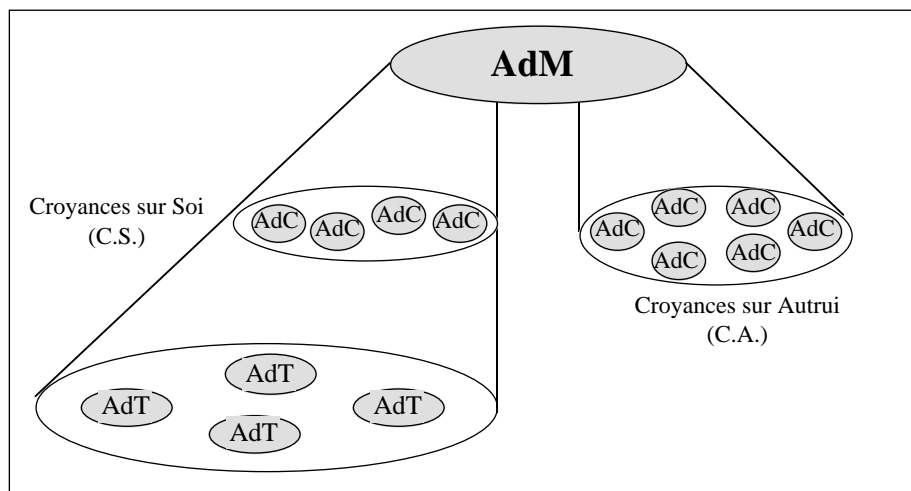


Figure 7.2. : Structure d'un agent de médiation

La structure d'un AdM est composée des cinq éléments définis au paragraphe 4.6.2.

Un AdM a pour **compétences** les compétences et les croyances qu'il possède sur les AdTs qu'il contient.

Un AdM possède un corpus de **croyances** sur les AdMs qu'il connaît. Cela permet d'accéder, dès le début du fonctionnement du système, à un certain nombre de services. A sa création, un

AdM est initialisé avec un corpus de croyances sur d'autres AdMs déjà présents. D'autres croyances sont acquises au fur et à mesure des interactions qu'il a avec ses congénères. Les AdMs ne possèdent qu'une vue locale à double titre :

- ils ont des croyances sur d'autres AdMs mais ne peuvent en aucun cas détenir de croyances sur des AdTs contenus dans les AdMs qu'ils connaissent.

- ils ne connaissent pas nécessairement l'existence de tous les autres AdMs composant le système.

Les AdMs possèdent plusieurs **aptitudes** parmi lesquelles, interpréter un message, apprendre en fonction des interactions ayant eu lieu, s'adapter aux perturbations en provenance de leur environnement...

Les AdMs **communiquent** par envoi de messages. Les messages échangés par les AdMs sont appelés des contextes d'AdT. Un contexte d'AdT est un AdT privé de toutes les aptitudes communes à tous les agents, telles que la fonction d'interprétation standard, les attitudes sociales, les fonctions de communication... La communication inter-AdMs entraîne implicitement la migration d'un AdT sur l'AdM avec qui l'AdM courant interagit,

Enfin, un AdM a comme **attitude sociale** la **coopération**. Le comportement coopératif des agents se caractérise par la détection et le traitement de situations non coopératives en vue d'une réorganisation décidée de manière autonome et locale.

7.3. LA COMMUNICATION CHEZ L'ADM

Le contexte correspond à l'unité d'échange entre AdMs. Comme les agents communiquent par envoi de messages, nous parlerons également de message typé par un contexte. Le contexte correspond à la requête devant être résolue dans le cas d'une migration pour rechercher de l'aide sur un site distant ou à la réponse éventuelle, déterminée dans le cas d'un retour vers le site initial.

Un AdM contient de très nombreux AdTs et possède des croyances sur d'autres AdMs avec qui il échange des informations. Un AdM a la capacité d'analyser l'état des AdTs qu'il contient. Lorsqu'un AdM sait qu'un de ses AdTs est bloqué (ce dernier lui ayant signalé son état), il essaie de le mettre localement en relation avec un ou plusieurs AdT(s) susceptible(s) de l'aider. Si aucun n'est capable de supprimer le blocage, l'AdM va initier une communication avec un ou plusieurs AdM(s), qui, selon son point de vue, est (sont) susceptible(s) d'aider l'AdT bloqué. Il utilise pour cela la relaxation restreinte et son jugement est fondé sur les croyances qu'il possède sur les AdMs qu'il connaît. Il s'ensuit la migration d'un clone de l'AdT vers un ou plusieurs AdMs qu'il juge localement pertinent(s).

Comme l'illustre la figure 7.3., l'AdM peut recevoir un message en provenance de trois sources émettrices c'est-à-dire d'un autre AdM (les cas ❶ et ❷ sont distingués car le premier fait référence à un message non signé alors que le second fait référence à un message signé -communication directe, réponse-), d'un AdT (cas ❸) ou d'un timer (cas ❹).

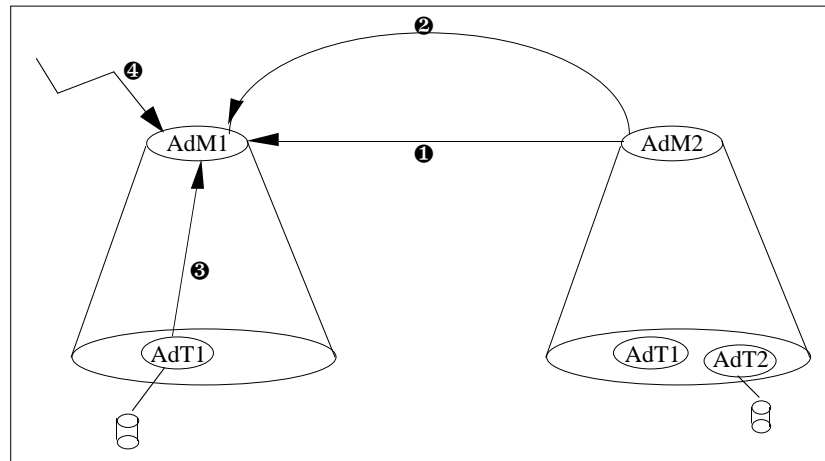


Figure 7.3. : Les différents interlocuteurs d'un AdM

Nous allons à présent sommairement résumer le comportement d'un AdM lors de la réception d'un message (pour plus de précisions vous pouvez consulter [Camps, 1997b]). L'interprétation est supposée être effectuée dès la réception du message.

7.3.1. Message en provenance d'un AdM (cas ❶ et ❷)

Un AdM peut recevoir trois types de message en provenance d'un de ses congénères : soit une requête à résoudre ou un message reçu par communication spontanée (cas ❶ sur la figure 7.3.1.), soit enfin une réponse à une requête qu'il n'a pas pu résoudre localement (cas ❷).

Dans le cas où le message reçu est une demande de résolution :

- si l'AdM se juge compétent (cela signifie qu'il pense contenir un ou plusieurs AdT(s) compétent(s)), il agentifie le message, c'est-à-dire le transforme en AdT et lui fournit la ou les croyance(s) nécessaire(s) pour qu'il puisse interagir de manière autonome avec les AdTs qui sont susceptibles de l'aider. Il lui communique ensuite la requête à résoudre. L'AdM fait alors office d'adaptateur pour l'AdT nouvellement créé.

- si l'AdM se juge incompetent pour résoudre la requête, il va étudier les croyances qu'il possède sur les AdMs qu'il connaît. S'il juge certains d'entre eux compétents pour résoudre la requête, il la leur relaxe.

Lorsqu'un AdM reçoit spontanément une information d'un autre AdM, si un de ses AdTs semble intéressé par celle-ci il la lui communique, sinon il ne fait rien. Ce jugement s'effectue en étudiant son réseau de croyances sur ses compétences.

Enfin, dans le cas ②, l'AdM met alors à jour son réseau de croyances sur autrui puis communique la réponse à l'AdT effectivement demandeur.

7.3.2. Message en provenance du timer (cas ③)

Un timer est une horloge permettant de gérer le temps d'attente d'une réponse associée à une requête émise.

Lorsque le timer concerne une requête envoyée à un autre AdM (soit AdM2), cela signifie que AdM1 a suffisamment attendu la réponse concernant cette requête. Si AdM2 a répondu avant la fin du temps de fin d'attente de AdM1 (signifié par le timer) cela n'annule pas le temps d'attente de AdM1. En effet, bien que AdM2 ait répondu, il peut avoir relaxé la requête à d'autres agents, auquel cas AdM1 peut recevoir des réponses d'autres AdMs que AdM2.

Dès la réception du "timeout", l'AdM courant n'attend plus de réponse éventuelle à sa requête et agit en conséquence c'est-à-dire met à jour son réseau de croyances sur autrui en tenant compte du fait qu'il y ait eu réponse, si réponse il y a eu.

7.3.3. Message en provenance d'un AdT (cas ④)

Un AdM peut recevoir un message en provenance d'un de ses AdTs contenant :

- soit la mention "bloqué" (par le biais d'un '?' comme force illocutoire du message transmis) ce qui signifie que l'AdT n'a pas terminé de résoudre la requête qui lui a été assignée et qu'il ne peut plus rien faire localement sur l'AdM,

- soit le feed-back de l'usager exprimant la satisfaction de ce dernier suite à la réponse fournie ou encore le feed-back d'un AdT émissaire signalant à l'AdM qu'un AdT particulier lui a fourni une réponse et donc qu'il doit remigrer sur son site initial pour transmettre la réponse à l'AdT dont il est issu.

Dans le premier cas, si l'AdT bloqué est un agent représentant, l'AdM va essayer de le débloquent en regardant s'il peut le mettre en contact avec un AdT qu'il contient et qui est susceptible de pouvoir participer à la résolution. S'il ne possède pas localement un AdT pertinent, il va initier une communication avec un ou plusieurs AdM(s) susceptible(s) de faire avancer la résolution.

Si maintenant l'AdT bloqué est un agent émissaire (qui est encore bloqué malgré les croyances que l'AdM lui a déjà fournies), l'AdM va regarder s'il connaît des agents AdMs pertinents. Si tel est le cas, il initie une communication avec l(es) AdM(s) alors trouvé(s), sinon il retourne # (signifiant son ignorance) à l'AdTe qui se tue.

Dans le second cas, le comportement de l'AdM est différent suivant la valeur du feed-back.

- s'il est négatif, l'AdM va mettre à jour son réseau de croyances sur ses compétences et, s'il a déjà participé à la résolution, voir s'il connaît des AdM(s) compétent(s) qui pourraient intervenir.

- si le feed-back est positif, l'AdM met à jour son réseau de croyances sur ses compétences et, si l'émetteur du feed-back est un agent émissaire, l'AdM initie une communication inter-AdMs pour retourner la réponse à l'AdM demandeur (retour du clone sur son site initial).

7.4. L'INTERPRÉTATION CHEZ L'ADM

L'interprétation permet à l'AdM de savoir s'il est localement compétent pour traiter le message reçu. Pour cela, le module d'interprétation étudie les croyances que ce dernier possède sur les AdTs qu'il contient dans le but de rechercher des AdTs pertinents pour la requête. Le module d'interprétation étudie également les croyances sur les autres AdMs. Le module d'interprétation permet plus précisément de mettre en évidence les situations non coopératives définies dans le cadre de l'étude.

7.4.1. Instanciation des situations non coopératives

Trois des situations non coopératives définies au paragraphe 5.2. ont été instanciées dans le cadre des AdMs. Les situations d'improductivité et d'incompétence conduisant à un traitement identique pour revenir à une situation coopérative n'ont pas été distinguées et sont regroupées sous le nom d'incompétence totale. La situation d'inutilité est inexistante dans le sens où les messages qui transitent entre AdMs sont de la forme question/réponse : donc toute réponse déterminée correspond implicitement à une demande. L'aptitude des AdMs à raisonner sur autrui leur permet de détecter les situations de concurrence et de voir la complémentarité d'autrui par rapport à eux. Ce deuxième point correspond à ce que nous avons appelé la compétence partielle et n'est pas considéré comme un comportement coopératif. Enfin, la situation de conflit est inexistante car elle s'inscrit en dehors de la problématique des AdMs : ils ne peuvent pas par eux-mêmes détecter une situation conflictuelle. En effet, deux AdMs désirant accéder à un tiers pour y effectuer une transaction identique, ne sont pas capables de savoir, par eux-mêmes, que leur demande respective gêne l'autre. Seul le service accédé sera à même de mettre en évidence cette situation non coopérative. Nous avons alors :

- **incompétence totale** lorsque l'AdM ne peut associer aucun sens au message qu'il a reçu,

- **compétence partielle** lorsque seule une partie du message est incomprise par l'AdM,
- **ambiguïté** quand le message reçu a plusieurs sens pour l'AdM,
- **concurrence** lorsque deux AdMs veulent accéder au même site pensant y trouver un service particulier.

Nous allons maintenant présenter comment ces situations sont détectées.

7.4.2. Détection des situations non coopératives

Chaque AdM possède deux réseaux de croyances, un réseau de croyances sur ses propres compétences (noté C.S.) et un réseau de croyances sur les compétences d'autrui (noté C.A.).

Le réseau de croyances sur soi permet de fournir aucun, un ou plusieurs AdT(s), susceptible(s) de pouvoir participer à la résolution en cours. Il fournit également, pour chaque AdT, le ou les termes qui ont permis de le juger pertinent (il s'agit de Croyance sur la figure 7.4.2).

Le réseau de croyances sur autrui permet de fournir le ou les AdM(s) pertinent(s) pour la requête en cours de résolution (AdT bloqué localement) si toutefois il en existe. Ici encore, le réseau procure, pour chaque AdM, les termes ayant permis de le qualifier de compétent pour la requête en cours.

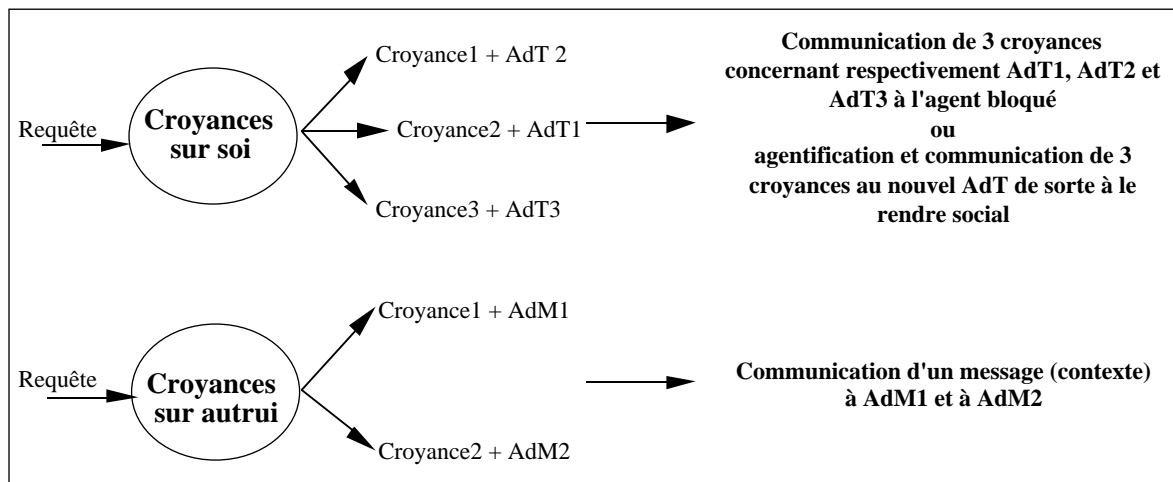


Figure 7.4.2. : Entrées/sorties des réseaux de croyances d'un AdM

Une fois la requête insérée dans C.A. et C.S., une table de croyances est construite pour chaque réseau à partir des résultats fournis. Il s'agit :

- d'une table de croyances portant sur les croyances que l'agent possède sur ses propres compétences appelée T.C.S.
- d'une table de croyances portant sur les croyances que possède l'agent sur autrui, appelée T.C.A.

7.4.2.1. Construction des deux tables de croyances

Chaque agent construit à la réception d'un message, une table de croyances sur soi (notée T.C.S.) et une table de croyances sur autrui (notée T.C.A.).

La table T.C.S. comporte trois champs :

- le nom de l'AdT (NomAdT) concerné par la croyance "Croyance" et pertinent pour la résolution en cours
- la liste des termes pertinents (Croyance) ayant permis de juger l'AdT "NomAdT" pertinent pour la requête en cours,
- le type de situation détectée, à savoir la compétence ou l'ambiguïté.

La table T.C.A. comporte également trois champs :

- le nom de ou des AdM(s) (NomAdM) susceptible(s) de pouvoir traiter le message interprété,
- la liste des termes pertinents (Croyance) ayant permis de juger l'AdM "NomAdM" pertinent pour la requête en cours,
- le type de situation détectée, à savoir la compétence partielle (complémentarité) ou la concurrence.

Nous allons à présent expliciter la mise en évidence des situations non coopératives.

7.4.2.2. Mise en évidence de l'incompétence et de l'ambiguïté

Les situations non coopératives telles que l'incompétence, l'ambiguïté sont locales à l'agent récepteur du message à interpréter. Elles peuvent être mises en évidence en exploitant les sorties du réseau de croyances sur soi.

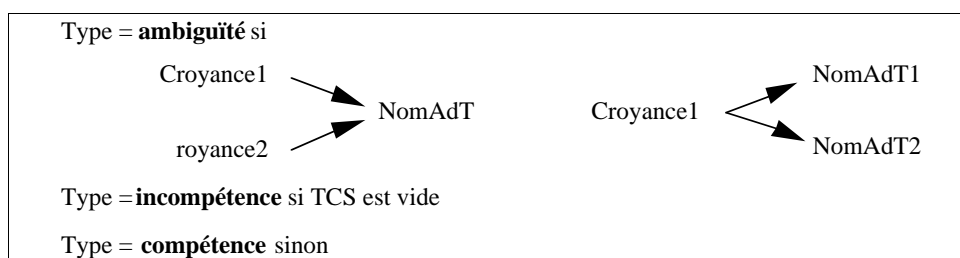


Figure 7.4.2.2. : Mise en évidence de l'ambiguïté/l'incompétence/la compétence

La situation sera qualifiée de situation d'**ambiguïté** dans deux cas :

- 1) si l'appariement entre 2 croyances différentes et associées à un même NomAdT est supérieur à 90%, autrement dit, si un agent particulier a été jugé pertinent pour deux raisons quasiment semblables mais toutefois distinctes,

2) si une même croyance est associée à deux NomAdTs différents, en d'autres termes si deux agents distincts ont été jugés pertinents pour les mêmes raisons.

Dans tous les autres cas, si le réseau fournit un résultat, l'agent est dit **compétent** pour traiter le message qui lui est envoyé.

Bien évidemment, si le réseau de croyances ne fournit aucun résultat cela reflète l'**incompétence** de l'agent récepteur.

7.4.2.3. L'incompétence partielle, la concurrence

La détection des situations non coopératives telles que la compétence partielle et la concurrence ne peuvent se faire que par rapport aux autres agents. Les sorties du réseau de croyances sur autrui sont donc exploitées pour les mettre en évidence.

Les différents types dans la table T.C.A. sont déterminés de la manière suivante : l'agent récepteur compare une à une chaque croyance de sa T.C.A. avec celles de sa T.C.S. en conservant le plus important degré d'appariement trouvé. Si ce degré est inférieur à une borne minimale, les agents résolvent des requêtes relativement différentes donc sont complémentaires. Dans le cas où le degré d'appariement est supérieur à une valeur maximum, les agents traitent des requêtes très similaires : ils sont concurrents. Entre ces deux valeurs de références, le rôle de l'agent est inconnu.

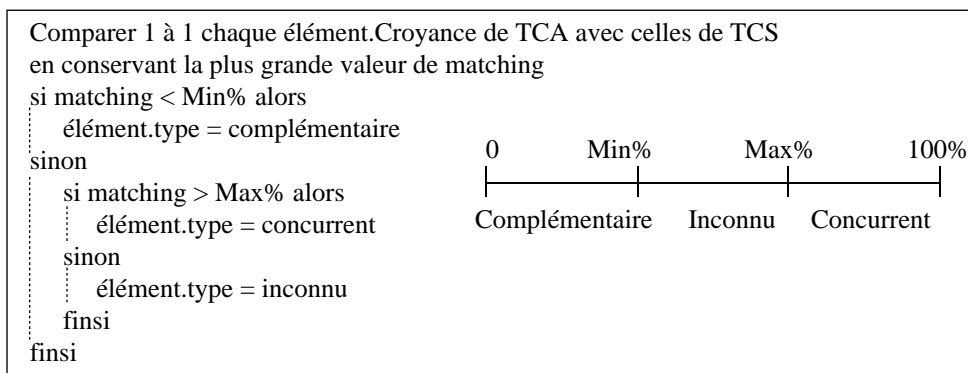


Figure 7.4.2.3. : Mise en évidence de la compétence partielle, la concurrence

Le type inconnu signifie que l'on ne sait pas déterminer si l'agent NomAdM est un agent complémentaire ou concurrent avec l'AdM courant.

Afin de pouvoir classer NomAdM dans l'une des deux catégories (complémentaire ou concurrent), l'AdM va réinjecter la croyance associée à NomAdM (obtenue après insertion de la requête dans le réseau de croyances sur autrui) dans son réseau de croyances sur soi. Si aucun résultat sort du

réseau de croyances sur soi cela signifie que NomAdM et l'AdM courant sont complémentaires. Si un ou plusieurs noms d'AdTs sont fournis par le réseau de croyances sur soi, quel que soit le type qui leur est associé, on dira que les agents sont concurrents.

7.4.3. Réactions face à ces situations non coopératives

Pour faire face à ces situations non coopératives, chaque acte individuel est simultanément guidé par l'objectif propre d'un agent et par la coopération avec les autres. La définition des actions à effectuer lorsqu'une situation non coopérative est détectée est fonction du champ expéditeur et du champ destinataire d'un message, le récepteur du message étant un AdM.

Le champ destinataire peut être vide dans le cas d'une localisation de service ou rempli lors d'une communication directe entre agents. De plus, un AdM ne peut recevoir un message qu'en provenance des trois sources énumérées au paragraphe 7.4. D'après les combinaisons possibles, quatre actions sont à définir.

7.4.3.1. Message sans destinataire en provenance d'un AdM (Action 1)

Dans le cadre d'une recherche, l'expéditeur du dit message ne peut être qu'un AdM. L'AdM récepteur a un comportement différent selon la force illocutoire véhiculée dans le message.

Les comportements (lorsque la force illocutoire est '?') sont listés dans le tableau de la figure 7.4.3.1. Les cases hachurées indiquent les cas qui ne peuvent pas survenir.

Comme le montre la figure 7.4.3.1., lorsqu'un agent reçoit un message pour lequel il n'est pas compétent (T.C.S. vide) et qu'il ne connaît personne susceptible de le comprendre (T.C.A. vide) l'agent ne fait rien (cas 1). L'émetteur du message se rendra compte, au bout d'un certain temps, qu'il ne reçoit pas de réponse de l'agent interrogé et mettra alors à jour ses croyances en conséquence.

Si en revanche (cas 8) un agent reçoit un message pour lequel il n'est pas compétent (T.C.S. vide) et qu'il connaît un ou plusieurs agents compétents (T.C.A. complémentaire) il le lui (leur) transmet en ne changeant rien de la structure du message : c'est la relaxation (cf. § 5.3.).

Lorsqu'un AdM reçoit un message pour lequel il est compétent, quel que soit le contenu de sa T.C.A. (cas 2, 4, 6, 8), il se crée l'objectif de résoudre le message reçu et l'agentifie, c'est-à-dire crée un nouvel AdT et lui communique les croyances concernant les AdTs susceptibles de l'aider dans sa résolution. Le nouvel AdT pourra ainsi interagir avec eux. Si, de plus, l'AdM connaît un ou plusieurs agents complémentaires (c'est le cas typique de la composition de services, cas 8), il change l'émetteur de la requête en la signant et l'envoie aux agents complémentaires.

Lorsqu'un AdM reçoit un message qu'il trouve ambigu et qu'il ne connaît personne de compétent ou qu'il connaît quelqu'un de concurrent (cas 3, 5, 7), il ne fait rien. Il n'essaie pas dans ce cas de lever l'ambiguïté, car le message ne lui est pas explicitement adressé ; il ne se risque pas à faire un travail inutile. De plus, il ne communique rien à l'expéditeur, ce dernier n'ayant pas besoin d'apprendre que l'AdT récepteur est en situation d'ambiguïté et ne sait pas comment traiter la requête. Dans le cas où un agent reçoit un message qu'il juge ambigu et qu'il connaît quelqu'un de complémentaire (cas 10), il le lui relaxe, c'est-à-dire le lui communique sans changer l'adresse de l'émetteur.

T.C.S.	Vide	Compétence	Ambiguïté
Vide	NOP (1)	Ajouter objectif Agentifier contexte (2)	NOP (3)
Concurrence		Ajouter objectif Agentifier contexte (4)	NOP (5)
Inconnu		Ajouter objectif Agentifier contexte Réinjecter Croyance de T.C.A. dans C.S. (6)	NOP (7)
Complémentaire	Relaxation (8)	Ajouter objectif Agentifier contexte (9) Communiquer contexte	Relaxation (10)

Figure 7.4.3.1. : Action à entreprendre lors de la réception d'une requête sans destinataire

Lorsque la force illocutoire est différente de '?' les comportements sont similaires à ceux décrits dans la figure 7.4.3.1. excepté dans deux cas :

- lorsque l'AdM est compétent, il ne se crée pas d'objectif parce que le message correspond à un transfert l'information et non à une sollicitation,
- lorsque l'AdM est compétent et qu'il connaît des AdMs qui lui sont complémentaires, au lieu de s'approprier égoïstement l'information qu'il vient de recevoir, il va, parce qu'il est coopératif, la relaxer à des congénères susceptibles d'être intéressés.

7.4.3.2. Message avec destinataire en provenance d'un AdM (Action 2)

Le message lui étant explicitement adressé, et s'agissant d'une réponse à une requête précédemment émise, l'AdM communique la réponse à l'AdT demandeur. Il met à jour son réseau

de croyances sur autrui car l'AdM avec qui il a communiqué lui a effectivement répondu. La pertinence de la réponse, qui pourra être appréciée par l'utilisateur, est prise en compte ultérieurement.

7.4.3.3. Message en provenance d'un AdT (Action 3)

Le comportement de l'AdM est celui décrit au paragraphe 7.3.3. Il diffère selon que le message correspond à une sollicitation ou à un feed-back d'un AdT et selon que l'AdT émetteur soit émissaire ou représentant.

7.4.3.4. Message en provenance du timer (Action 4)

Comme le message concerné par le timer ne peut être qu'un message communiqué à un autre AdM, l'AdM met à jour son réseau de croyances sur autrui. Il met également à jour sa table des objectifs en marquant l'action correspondante comme étant atteinte ou pas.

7.5. LE RAISONNEMENT CHEZ L'ADM

Le raisonnement est effectué au sein d'un AdM essentiellement par ses compétences c'est-à-dire par l'activation des AdTs pertinents en fonction de la requête en cours de résolution. Ce raisonnement peut aboutir à la construction d'AdTs ainsi qu'à la mise en relation d'AdTs pour résoudre efficacement le problème courant. Cette dernière action est mise en oeuvre en leur ajoutant de nouvelles croyances (fournies par le module d'interprétation).

Il permet également la gestion pertinente des objectifs de l'agent et des actions à entreprendre lors de la réception d'un "timeout", lui signalant la fin d'un délai d'attente associé à une requête particulière.

Le raisonnement chez l'AdM a pour but de transformer le contexte reçu en AdT c'est-à-dire d'agentifier ce contexte. Lorsqu'un AdM reçoit un message en provenance d'un autre AdM et qu'il se juge compétent, l'agentification de ce message est réalisée. Cette opération consiste à créer un AdT à partir du contexte reçu et des croyances susceptibles de lui permettre d'avancer dans la résolution (issues de son réseau de croyances sur soi (après insertion de la requête dans son réseau de compétences). Lorsque le nouvel AdT va interpréter le message reçu il va l'insérer dans son C.S. et dans son C.A. Les AdTs concernés par les croyances fournies seront alors jugés complémentaires ce qui permettra à l'AdT courant de leur transmettre respectivement la requête à résoudre.

7.6. RÉSULTATS OBTENUS A PARTIR DU SYSTÈME MULTI-AGENT D'ADMS

Les AdMs peuvent être qualifiés d'agents "cognitifs" dans le sens où leur comportement fait intervenir un certain raisonnement. Leurs comportements peuvent alors être analysés individuellement et nous nous attacherons à en étudier l'aspect coopératif. Les observateurs que nous sommes peuvent ainsi juger de l'intérêt global du comportement d'un agent, alors que lui seul l'a décidé d'après ses conséquences en termes coopératifs.

Le scénario sur lequel reposent les divers exemples est le suivant : AdT1 cherche à vendre une polo et désire trouver acquéreur, alors que AdT4 recherche à acheter une polo sous certaines conditions. Le but du scénario est de mettre en contact ces deux agents bien qu'ils n'ont aucune connaissance l'un de l'autre.

Les résultats obtenus illustrent la communication inter-AdMs ainsi que la communication spontanée au niveau des AdMs.

7.6.1. Mise en évidence de la communication inter-AdMs

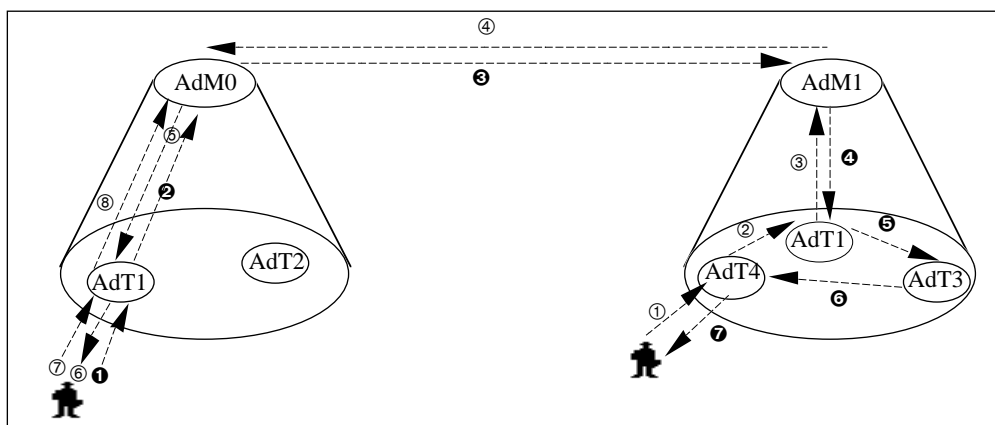


Figure 7.6.1. Communication inter-AdMs (réalisée lors des communication ③ et ④)

Contexte : le tableau ci-après résume les résultats obtenus suite à l'insertion de req1 émise par AdT1 dans les deux réseaux de croyances possédés par chaque agent constituant ce prototype.

	Réseau de croyances sur soi	Réseau de croyances sur autrui
AdM0	∅	AdM1
AdT1	∅	∅
AdT2	∅	∅
AdM1	AdT3	∅
AdT3	∅	AdT4
AdT4	AdT4	∅

Fonctionnement : AdT1 reçoit une requête en provenance de son adaptateur ❶. Ne connaissant aucun AdT compétent, il se trouve dans l'état bloqué et demande de l'aide à son AdM ❷.

AdM0 reçoit la requête de AdT1 ❷, l'interprète et ne se juge pas compétent : il connaît en revanche AdM1 qui semble pertinent. Il lui envoie donc la requête de AdT1 en l'ayant au préalable signée de son nom ❸.

AdM1 reçoit la requête de AdM0 ❸ et l'interprète. Il se juge compétent : l'AdT3 semble pertinent. Il met à jour son réseau de croyances sur autrui et agentifie le contexte reçu, en créant un nouvel AdT (appelé AdT1 sur la figure 7.6.1.). L'agentification consiste à créer un nouvel AdT (AdT1) et à lui fournir la ou les croyance(s) nécessaire(s) pour qu'il puisse interagir de manière autonome. AdM1 fait office d'adaptateur pour AdT1. Une fois l'agentification terminée, AdM1 communique la requête à résoudre à AdT1 ❹.

AdT1 interprète la requête en provenance de son adaptateur. Il juge AdT3 pertinent et lui envoie donc la requête ❺.

AdT3 reçoit la requête en provenance de AdT1 ❺, l'interprète et se juge incompetent. En revanche il connaît AdT4 qui paraît pertinent. AdT3 relaxe donc la requête à AdT4 ❻ après avoir mis à jour son réseau de croyances sur autrui.

AdT4 reçoit la requête en provenance de AdT1 ❻ (qui a été relaxée par AdT3). Il l'interprète et met à jour son réseau de croyances sur autrui. Il se trouve compétent et ne connaît personne de complémentaire. Il transmet donc la requête à son adaptateur ❼.

L'adaptateur de AdT4 transmet à AdT4 son désir d'acheter la voiture proposée par AdT1 ❶.

AdT4 retourne la réponse à AdT1 ❷ qui la communique à son adaptateur (AdM1) ❸. AdM1 met à jour son réseau de croyances sur soi et, grâce à sa table des objectifs, sait que la réponse reçue de AdT1 correspond à une requête de AdM0. AdM1 la retransmet alors à AdM0 ❹.

AdM0, grâce à sa table des objectifs sait qu'il s'agit d'une réponse à la requête de AdT1 et la lui transmet ❺. Il met également à jour son réseau de croyances sur autrui.

AdT1 transmet alors la réponse à son adaptateur ❻. Ce dernier lui communique son feed-back ❼.

AdT1 met alors à jour son réseau de croyance sur autrui et transmet le feed-back de son usager à AdM0 ❸ pour que ce dernier mette à jour à la fois son réseau de croyances sur soi et son réseau de croyances sur autrui.

7.6.2. Mise en évidence de la communication spontanée

Contexte : il est identique à celui fourni au paragraphe 7.6.1.

Fonctionnement : supposons que le scénario explicité au paragraphe 7.6.1. soit réalisé. AdT1 (situé sur AdM0) a reçu un message de AdT4 (situé sur AdM1) suite à une communication inter-AdMs.

Supposons que AdT4 se voit contraint de revenir sur sa décision. AdT4 reçoit alors de son adaptateur un message annulant l'offre précédemment émise. AdT4 transmet spontanément l'information aux agents susceptibles d'être intéressés, en l'occurrence AdT1. Deux cas peuvent alors se présenter :

- soit AdT1 est toujours présent sur AdM1. Il reçoit alors l'information de AdT4 et la transmet à son adaptateur (AdM1) qui la communique à AdM0.

- soit un certain temps s'est écoulé entre les deux transactions et AdT1 s'est supprimé suite à un "timeout". Dans ce cas là, la communication ne peut aboutir et l'AdT4 transmet la requête à AdM1 qui va communiquer l'information à AdM0 car son réseau sur autrui a encore trace de l'intérêt de AdM0 sur la vente d'une polo.

Le scénario continue ensuite comme précédemment.

7.6.3. Mise en évidence de la relaxation restreinte

Nous allons mettre en évidence dans ce paragraphe la mise en oeuvre de la relaxation restreinte au niveau des AdMs. La trace de cet exemple est fournie en annexe C.

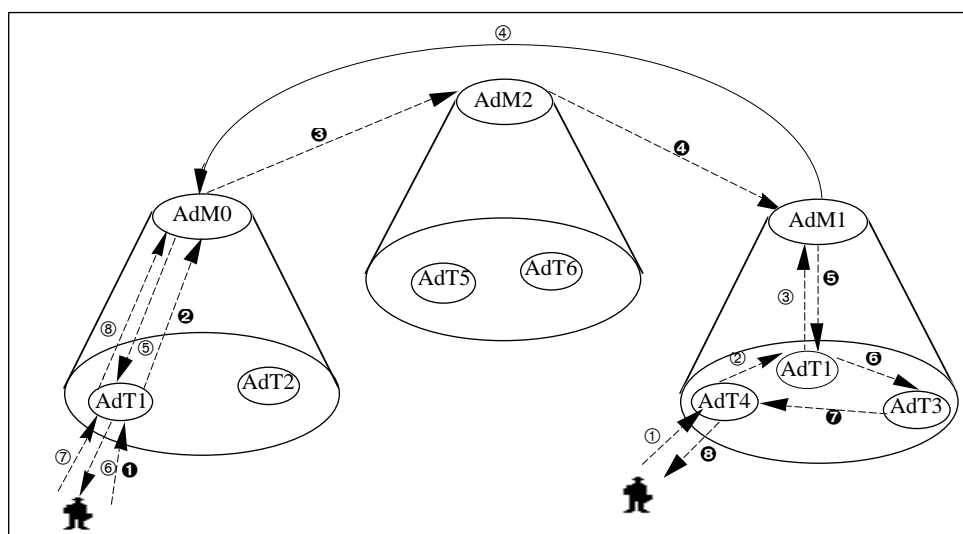


Figure 7.6.3. : Mise en évidence de la relaxation restreinte au niveau des AdMs (cas 4)

Contexte : nous supposons pour cela un système composé de 3 AdMs (AdM0, AdM1, AdM2) comprenant respectivement un certain nombre d'AdTs. Les résultats obtenus suite à l'insertion de req1 émise par AdT1 dans les deux réseaux de croyances possédés par chaque agent constituant ce prototype sont identiques à ceux fournis au paragraphe 7.6.1. AdM2 quand à lui, n'est pas compétent pour la requête en cours mais connaît AdM1, qui lui, le semble.

Fonctionnement : AdT1 reçoit une requête en provenance de son adaptateur ❶. Ne connaissant aucun AdT compétent, il se trouve dans l'état bloqué et demande de l'aide à son AdM ❷.

AdM0 reçoit la requête de AdT1 ❷, l'interprète ne se juge pas compétent : il connaît en revanche AdM2 qui paraît pertinent. Il lui envoie donc la requête de AdT1 ❸ en l'ayant au préalable signée de son nom.

AdM2 reçoit la requête de AdM0 ❸ et l'interprète. Il se juge incompetent mais connaît AdM1 qui lui est complémentaire. AdM1 relaxe la requête en provenance de AdM0 à AdM1 ❹.

AdM1 reçoit la requête de AdM0 ❹ (qui en fait provient d'une relaxation de AdM2). Il se juge compétent car l'AdT3 semble pertinent. Il met à jour son réseau de croyances sur autrui (AdM0 est intéressé par ...). Il agentifie le contexte reçu, en créant un nouvel AdT (appelé AdT1 sur la figure 7.6.3)...

La suite du scénario est identique à celui du paragraphe 7.6.1.

Lorsque AdT1 communique la réponse à son adaptateur ❸, AdM1 renvoie la réponse à AdM0 ❹ car c'est lui qui est à l'origine de la demande (table des objectifs). L'apprentissage est ensuite identique au cas précédent.

7.7. DISCUSSION

Le système multi-agent de médiation, représente le niveau le plus élevé de l'architecture d'ARCADIA. Le but d'un AdM est d'être en situation coopérative avec les AdMs qu'il connaît (son environnement) et les AdTs qu'il contient (son milieu intérieur). Les AdMs interagissent en fonction des croyances qu'ils possèdent les uns sur les autres.

La première remarque qui peut alors venir à l'esprit est que comme les croyances évoluent au fur et à mesure de la qualité des interactions ayant eu lieu entre AdMs et qu'elles peuvent être amenées à disparaître dans le cas où leur utilité n'est pas justifiée, la connexité entre AdMs peut être perdue. Pour éviter l'exil définitif d'un AdM, on peut supposer avoir un supra-AdM tenant le même rôle qu'un AdM classique vis à vis des AdTs qu'il contient, c'est-à-dire connaissant tous les AdMs constituant ARCADIA. Il mettrait alors en relation un AdM bloqué avec un ou plusieurs autres AdMs susceptibles de pouvoir résoudre le problème.

Peut être est-ce à cause du mot médiation qui le compose, qu'un AdM est parfois assimilé à un médiateur et donc à un tableau noir. Cet amalgame n'a pas lieu d'être pour plusieurs raisons :

- tout d'abord un AdM est un agent à part entière : il a ses propres compétences, ses propres croyances qui évoluent dynamiquement en fonction des interactions qu'il entretient,
- d'autre part, les compétences qu'il possède ne sont pas forcément exactes et ne sont pas le reflet exact des compétences des AdTs qui le composent.
- enfin, un tableau noir est une structure de donnée commune et passive que les agents consultent alors que les AdMs sont actifs et se comportent de manière coopérative.

D'autre part, dans l'exposé concernant le système multi-agent de médiation, il est possible de distinguer deux sortes d'apprentissage. Le premier concerne la mise à jour du réseau de croyances sur les compétences de l'agent de médiation suite au feed-back de l'utilisateur (resp. service), feed-back fourni par l'AdT représentant l'utilisateur (resp. service) et contenant l'appréciation de ce dernier quant à la satisfaction de la réponse fournie. Le second apprentissage concerne la mise à jour du réseau de croyances sur autrui de l'AdM lors d'une réponse fournie par un autre AdM. L'apprentissage prend uniquement en compte, cette fois-ci, le fait que l'AdM ait répondu ; il ne considère pas l'appréciation de l'utilisateur (resp. service) sur la qualité de la réponse. Celle-ci ne sera prise en compte que lorsque l'utilisateur (resp. service), ayant pris connaissance de la réponse, donnera son appréciation à son agent représentant qui, lui, le communiquera enfin à l'AdM.

Une autre remarque très importante, peut être plus adaptée aux agents de granularité inférieure, mais également intéressante à ce niveau : lorsqu'un nouvel AdM est créé par France Télécom, il est encapsulé avec les comportements coopératifs que nous préconisons. En aucun cas il ne peut avoir un comportement malhonnête visant à perturber le reste de la société. Si toutefois un agent malhonnête arrivait à s'immiscer dans la société, les croyances le concernant se réajusteraient au fur et à mesure des interactions effectives jusqu'à modéliser la vraie image de ce dernier et non celle qu'il prétendait avoir initialement. Malgré cet aparté, la caractéristique importante à retenir est que la coopération est une attitude sociale rémanente dont sont dotés tous les AdMs par encapsulation. Elle est nécessaire pour avoir un système multi-agent coopératif et donc un système fonctionnellement adéquat (cf. partie 2).

Enfin, nous avons mis en évidence, au travers de divers exemples, certains comportements coopératifs possédés par les AdMs. Ils sont à la base de l'auto-organisation du système car ils entraînent des modifications de liens entre agents en fonction des interactions vécues. Ainsi si un agent doit résoudre une requête pour laquelle il n'est pas compétent il va bénéficier de l'aide d'agents pertinents. Cette demande peut entraîner, en particulier dans le cas d'une relaxation restreinte, de nouveaux liens d'interaction. Si peu de temps après, le même agent est sollicité pour résoudre une requête similaire, la situation non coopérative d'incompétence ne sera plus détectée. L'agent aura appris et mis à jour son réseau de croyances sur autrui : le temps de traitement s'en trouvera nettement amélioré.

Nous avons également pu remarquer que bien que les AdMs poursuivent l'unique but d'être coopératifs avec les AdTs qu'ils contiennent et les autres AdMs, ils participent efficacement à la localisation du service ou de l'utilisateur recherché. Ils n'ont pas conscience de prendre partie à la recherche étant donné que ce n'est pas leur finalité. Pourtant, ils permettent de localiser le service ou l'utilisateur permettant de répondre au problème en cours de résolution sur un site particulier.

CHAPITRE 8

LE SYSTÈME MULTI-AGENT D'AGENTS DE TRANSACTION

8.1. INTRODUCTION

Les Agents de Transaction (AdTs) se distinguent des AdMs par leur dynamique beaucoup plus grande en terme de création et de suppression.

Les AdTs sont sédentaires (agents représentants) ou mobiles (agents émissaires) suivant leur rôle. Ils peuvent migrer entre AdMs selon leur besoin ce qui revient à les considérer comme des agents mobiles pour un observateur.

Les AdTs contiennent et manipulent la connaissance des usagers et services qu'ils représentent.

Un AdT peut être créé dans trois cas : soit à la création d'un nouveau service, soit à la première connexion d'un usager ou soit enfin lorsqu'un AdT migre pour trouver une réponse à la requête qu'il ne peut satisfaire dans le site où il se trouve. Cette création a pour effet de doter l'agent d'aptitudes communes à tous.

8.2. STRUCTURE D'UN ADT

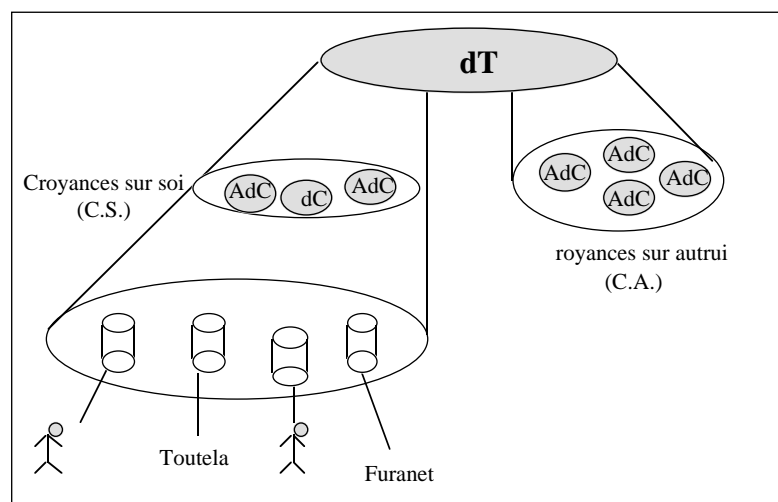


Figure 8.2. Structure d'un AdT

Qu'il soit sédentaire ou migrant, un AdT est composé des cinq éléments définis au paragraphe 4.6.2. :

Un AdT a pour **compétence** les images opératoires de l'utilisateur ou du service qu'il représente.

Un AdT possède un corpus de **croyances** sur les compétences d'autres AdTs présents sur le même site que lui. Ces croyances évoluent au fur et à mesure des interactions qu'il a avec ses congénères. Il n'a qu'une vue locale et ne détient des croyances que sur les AdTs situés sur le même AdM que lui.

Les AdTs possèdent plusieurs **aptitudes** parmi lesquelles, interpréter un message, apprendre en fonction des interactions ayant eu lieu, s'adapter aux perturbations en provenance de leur environnement...

Les AdTs **communiquent** par envoi de messages. Un message est une information sur le domaine correspondant à la réalisation d'une transaction.

Enfin, un AdT a comme **attitude sociale la coopération**. Le comportement coopératif des agents se caractérise par la détection et le traitement de situations non coopératives en vue d'une réorganisation décidée de manière autonome et locale.

8.3. LA COMMUNICATION CHEZ L'ADT

La requête ou la réponse correspond à l'unité d'échange entre les agents de transaction. Parce que nos agents communiquent par envoi de messages, nous parlerons également de message typé par une requête ou une réponse.

Comme l'illustre la figure 8.3., l'AdT peut recevoir un message en provenance de quatre sources émettrices c'est-à-dire de l'AdM dans lequel il est situé (cas ❶), d'un autre AdT (cas ❷), de son adaptateur (cas ❸) ou d'un timer (cas ❹).

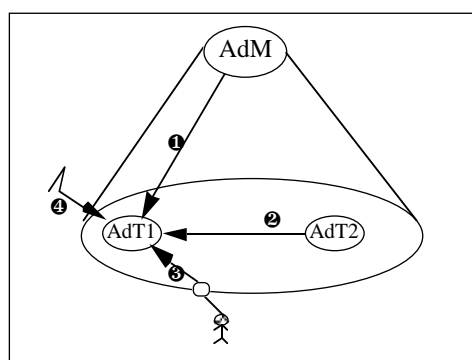


Figure 8.3. : Les différents interlocuteurs d'un AdT

Nous allons à présent brièvement présenter le comportement d'un AdT lors de la réception d'un message (pour plus de détails, vous pouvez vous reporter à [Camps, 1997b]). L'interprétation est supposée être effectuée dès la réception d'un message.

8.3.1. Message en provenance d'un AdM (cas ①)

Suite à une sollicitation de sa part (parce qu'il est bloqué), un AdT peut recevoir de son AdM une croyance (ou plusieurs) lui permettant d'établir une communication avec un congénère susceptible de lui permettre d'avancer dans la résolution de son problème. Il met alors à jour son réseau de croyances sur autrui en insérant la croyance reçue puis essaie de nouveau de résoudre la requête. Si l'AdM ne peut débloquent un AdT qu'il contient, il lui retourne la requête avec # comme force illocutoire (signifiant son incapacité). Deux situations peuvent alors se présenter : soit l'AdT est un agent représentant auquel cas il fait part de la réponse de l'AdM à son adaptateur, soit l'AdT est un agent émissaire et il se supprime.

Une communication peut également avoir lieu entre un AdM et un de ses AdTs lorsque l'AdM reçoit une réponse à une demande qu'il a propagée vers d'autres AdMs. Dans ce cas, l'AdM communique directement la réponse à qui de droit, c'est-à-dire à l'AdT qui, chez lui est à l'origine de la demande. L'AdT doit alors transmettre la réponse à son adaptateur et également mettre à jour sa table des objectifs.

8.3.2. Message en provenance d'un timer (cas ④)

Une horloge gère le temps d'attente d'une réponse associée à une requête émise. Si le timer concerne un message envoyé à un AdT, cela veut dire que l'AdT ne doit plus attendre une réponse mais au contraire mettre à jour son réseau de croyances sur autrui en conséquence, c'est-à-dire prendre en compte l'incapacité de l'AdT sollicité. Si, de plus, l'AdT recevant le timer est un agent émissaire, il se supprime.

Lorsque le timer concerne une requête transmise à son adaptateur, l'AdT met à jour son réseau de compétences.

8.3.3. Message en provenance d'un AdT (cas ②)

Un AdT peut recevoir une requête à résoudre, auquel cas, si à l'issue de l'interprétation, il se juge compétent, il la transmet à son adaptateur pour résolution après avoir inséré un nouvel objectif dans sa table des objectifs. Dans le cas contraire, il la relâche à des congénères compétents s'il en connaît. S'il ne connaît pas d'AdT compétent et si la requête lui est explicitement adressé, il sollicite l'AdM dans lequel il est situé.

Un AdT peut également recevoir une réponse à une requête précédemment émise, auquel cas il met à jour son réseau C.S. et la transmet à son adaptateur.

Enfin, un AdT peut recevoir un message envoyé spontanément par un autre AdT situé sur le même site. S'il est intéressé par l'information reçue, il la transmet à son adaptateur. Dans le cas contraire, il la relâche à des AdTs susceptibles d'être intéressés.

8.3.4. Message en provenance de son adaptateur (cas ③)

Trois types de message peuvent parvenir à un AdT de la part de son adaptateur. Il peut s'agir :

- soit d'une requête à résoudre, auquel cas l'agent l'interprète. Il la transmet à des AdTs compétents s'il en connaît. Dans le cas contraire, il sollicite l'aide de son AdM.

- soit d'un feed-back de l'utilisateur ou du service représenté par l'AdT, lui signifiant le jugement de ce dernier quant à la qualité de la réponse fournie. L'AdT met alors à jour son réseau de croyances (sur ses compétences, si le feed-back ne concerne pas un AdT particulier, sur autrui s'il concerne au contraire un autre AdT) et communique le feed-back reçu à l'AdM auquel il appartient pour que celui-ci mette lui aussi à jour son réseau de croyances sur ses compétences.

- soit enfin d'une réponse à une sollicitation antérieure, auquel cas il la transmet à qui de droit.

8.4. L'INTERPRÉTATION CHEZ L'ADT

L'interprétation permet à un AdT de savoir s'il est localement compétent pour traiter le message reçu et s'il connaît des AdTs pertinents. Le module analyse pour cela les réseaux de croyances de l'AdT sur ses propres compétences et sur autrui. Le module d'interprétation permet plus précisément de mettre en évidence les situations non coopératives définies dans le cadre de l'étude.

8.4.1. Instanciation des situations non coopératives

Pour les mêmes raisons que celles énoncées au paragraphe 7.5.1, quatre situations non coopératives sont distinguées à ce niveau. Nous identifions donc:

- l'**incompétence totale** lorsque l'AdT ne peut associer aucun sens au message qu'il a reçu,
- la **compétence partielle** lorsque seule une partie du message est incomprise par l'AdT,
- l'**ambiguïté** quand le message reçu a plusieurs sens pour l'AdT,
- la **concurrence** lorsque deux AdTs veulent accéder à une ressource limitée et leur demande excède l'offre.

Nous allons maintenant présenter comment ces situations sont détectées.

8.4.2. Détection des situations non coopératives

Chaque AdT, quel que soit son type possède deux réseaux de croyances. Tout d'abord un réseau de croyances sur soi (C.S.) qui admet en entrée une requête c'est-à-dire un message en provenance d'un autre AdT, de l'adaptateur de l'AdT ou de l'AdM dans lequel il est situé. Il fournit en sortie les champs de l'interface qui chez lui seront remplis lors de l'interrogation ainsi que les termes (appelés "croyances" sur la figure 8.4.2.) qui semblent pertinents pour l'interrogation.

Un AdT possède ensuite un réseau de croyances sur les autres (C.A.). Il admet en entrée une requête c'est-à-dire un message en provenance d'un autre AdT. Il fournit en sortie le ou les AdT(s) qui selon son point de vue sont capables de l'aider à résoudre la requête saisie en entrée.

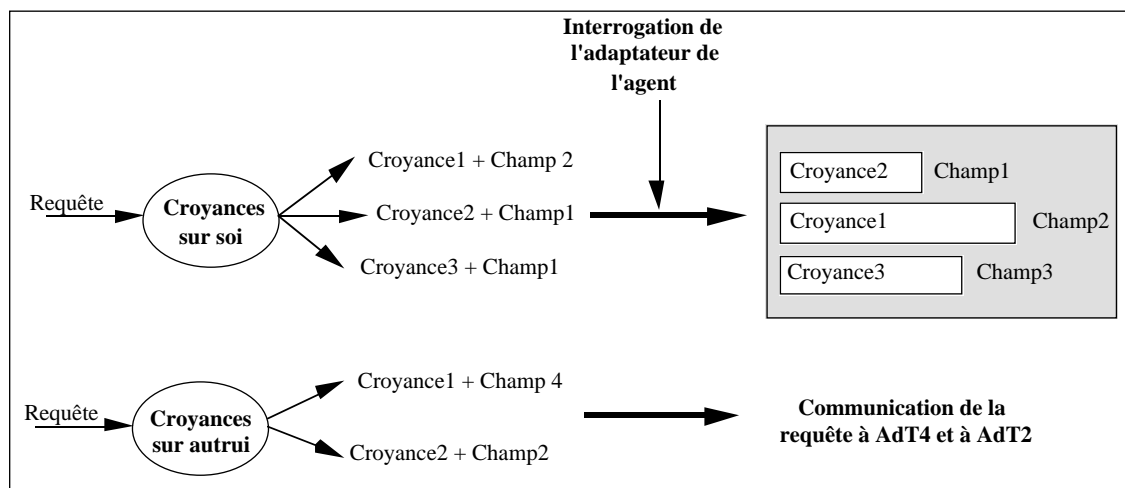


Figure 8.4.2. : Entrées/sorties des réseaux de croyances d'un AdT

Après avoir inséré la requête dans C.S. et C.A., une table de croyances est construite pour chaque réseau. On distingue :

- la table de croyances portant sur les croyances que l'AdT possède sur ses propres capacités (T.C.A.).
- la table de croyances portant sur les croyances que possède l'AdT sur d'autres AdTs (T.C.S.).

Nous allons à présent voir leur mode de construction.

8.4.2.1. Construction des deux tables de croyances

Chaque AdT construit à la réception d'un message, une table de croyances sur soi et une table de croyances sur autrui .

La table de croyances T.C.S. comporte trois champs :

- le nom du champ de l'interface (Champ) qui permet de dialoguer avec le service représenté par l'AdT et dans lequel doit être inscrite la croyance,
- la liste des termes pertinents (Croyance) pour interroger l'adaptateur,
- et le type de situation détectée, à savoir la compétence ou l'ambiguïté.

La table T.C.A. comporte également trois champs :

- le nom du ou des AdT(s) (NomAdT) susceptible(s) de pouvoir traiter le message interprété,
- la liste des termes (croyance) ayant permis de qualifier NomAdT de pertinent,
- le type de situation détectée, à savoir la compétence partielle (complémentarité) ou la concurrence.

8.4.2.2. L'incompétence, l'ambiguïté

L'incompétence ou l'ambiguïté sont locales à l'agent récepteur du message à interpréter. Elles peuvent être mises en évidence en exploitant les sorties du réseau de croyances sur soi.

Un AdT trouve un message ambigu dans deux cas :

- si deux listes de termes différentes (croyances) peuvent s'insérer dans un même champ de l'interface de l'adaptateur lors de l'interrogation,
- ou lorsqu'une même liste de termes (croyance) peut s'inscrire dans deux champs différents de l'interface de l'adaptateur.

Dans tous les autres cas, l'agent est dit compétent pour traiter le message qui lui est envoyé.

Bien évidemment, si le réseau de croyances ne fournit aucun résultat cela reflète l'incompétence de l'agent.

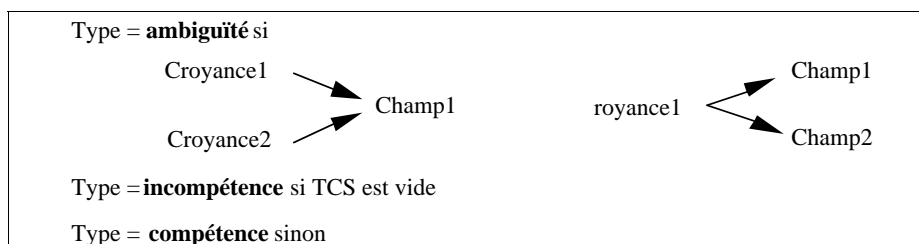


Figure 8.4.2.2. : Mise en évidence de l'ambiguïté/l'incompétence/la compétence

8.4.2.3. La compétence partielle, la concurrence

La détection des situations non coopératives telles que la compétence partielle, la concurrence ou le conflit sont mises en évidence grâce aux sorties du réseau de croyances sur autrui.

Les différents types dans la table T.C.A. sont déterminés exactement de la même manière que pour l'AdM (cf. paragraphe 7.5.2.3. en remplaçant NomAdM par NomAdT).

8.4.3. Réactions face à ces situations non coopératives

La définition des actions à effectuer lorsqu'une situation non coopérative est détectée est fonction, ici aussi, du champ expéditeur et du champ destinataire d'un message.

Un AdT ne peut recevoir un message qu'en provenance des quatre sources énumérées au paragraphe 8.4. D'après les combinaisons possibles, cinq actions sont à définir.

8.4.3.1. Message sans destinataire en provenance d'un autre AdT (Action 1)

Dans le cadre d'une recherche, c'est-à-dire lorsque le champ destinataire logique du message est vide, l'expéditeur du dit message peut être un AdM ou un autre AdT. Lorsque l'expéditeur est un AdT, l'AdT récepteur a un comportement différent selon la force illocutoire véhiculée dans le message.

Les comportements (lorsque la force illocutoire est '?') sont identiques à ceux présentés dans la table 7.4.3.1., à ceci près que, lors d'une compréhension, il n'y a pas d'agentification du contexte mais communication du message reçu à l'adaptateur de l'AdT courant.

8.4.3.2. Message en provenance d'un AdT avec destinataire (Action 2)

Dans le cas où l'expéditeur est un autre AdT, l'AdT interprète le message qui lui a été envoyé et donc construit T.C.S. et T.C.A.

Le message lui étant explicitement adressé, même s'il ne le comprend pas, il le communique à son adaptateur pour raison de confidentialité. L'interface permettant cet échange est supposée posséder une zone dans laquelle peut être inscrit tout type de message.

Si de plus le message communiqué possède '?' comme force illocutoire, l'agent se rajoute un objectif, celui de le résoudre.

8.4.3.3. Message en provenance d'un AdM (Action 3)

Dans ce cas, il peut s'agir d'une réponse d'un AdM suite à une sollicitation de l'AdT qui était en situation de blocage. Le message transmis par l'AdM est une croyance sur un AdT qu'il contient et qui, selon son point de vue, peut aider l'agent bloqué. L'AdM détermine la croyance à transmettre en injectant la requête de l'agent demandeur dans son réseau C.S. et communique alors à l'AdT

bloqué la croyance issue du réseau ainsi que le nom de l'AdT pertinent, ce dernier étant inscrit dans le champ émetteur logique (cf. § 6.5.1.1.). Suite à la réception de la croyance, l'AdT récepteur bloqué, va mettre à jour son réseau de C.A. La mise à jour consiste à injecter la croyance dans son réseau de croyances. Une fois ceci effectué, il réinjecte la requête initiale dans C.A. et soumet ensuite la requête aux agents pertinents ainsi trouvés après avoir mis en évidence d'éventuelles situations non coopératives grâce à T.C.A.

Il peut également s'agir d'une réponse à une requête précédemment émise par l'AdT (suite à un blocage) qui a été résolue sur un site distant. L'AdT va alors transmettre à son adaptateur la réponse que lui a communiquée par l'AdM après avoir mis à jour sa table des objectifs.

8.4.3.4. Message en provenance du timer(Action 4)

Dans ce cas, l'agent va avoir un comportement différent selon que le timer concerne une requête transmise à son adaptateur ou à un autre AdT. Dans le premier cas, l'AdT met à jour son réseau de croyances sur ses propres compétences. Il saura ainsi que l'utilisateur ou le service qu'il représente n'est pas compétent pour la requête transmise. Dans le second cas, il met à jour son réseau de croyances sur autrui.

Enfin, dans les deux cas, l'AdT met à jour sa table des objectifs en conséquence : cela consiste à marquer le but comme atteint ou pas.

8.4.3.5. Message en provenance de l'adaptateur(Action 5)

Lorsque l'expéditeur du message est un adaptateur, l'AdT regarde dans un premier temps s'il s'agit d'une demande, d'une réponse, d'une information transmise spontanément ou d'un feed-back.

S'il s'agit d'une demande, l'AdT regarde s'il s'agit d'une demande ciblée directe ou pas. Si ce n'est pas le cas, il étudie ses croyances sur autrui pour voir s'il connaît un ou plusieurs AdTs pertinents à qui transmettre la requête. S'il n'en connaît pas, il sollicite son AdM.

S'il s'agit d'une réponse à une requête précédemment transmise, il met à jour son réseau de croyances sur soi puis communique la réponse à l'agent demandeur.

S'il s'agit d'une information communiquée spontanément, l'AdT étudie ses croyances sur autrui pour la transmettre aux AdTs qu'il connaît et qui peuvent être intéressés.

S'il s'agit d'un feed-back, il met à jour son réseau de croyances sur soi ou sur autrui selon que l'agent concerné est un AdT situé sur le même site ou pas. Dans les deux cas, l'AdT transmet le feed-back à son AdM afin que ce dernier mette également à jour ses croyances.

8.5. RÉSULTATS OBTENUS A PARTIR DU SYSTÈME MULTI-AGENT D'ADTS

Les AdTs peuvent être qualifiés d'agents "cognitifs" dans le sens où leur comportement fait intervenir un certain raisonnement. Leurs comportements peuvent alors être analysés individuellement et nous nous attacherons à en étudier l'aspect coopératif. Les observateurs que nous sommes peuvent ainsi juger de l'intérêt global du comportement d'un agent, alors que lui seul l'a décidé d'après ses conséquences en termes coopératifs.

Pour étudier leur comportement, nous avons fait l'hypothèse que tous les AdTs se trouvent dans le giron d'un seul agent de médiation (AdM0). Nous supposons un système comprenant sept AdTs parmi lesquels deux représentent respectivement un usager (AdT1 et AdT4) et cinq représentent respectivement un service parmi ceux énoncés au paragraphe 6.5.2.

8.5.1. Mise en évidence de la relaxation restreinte

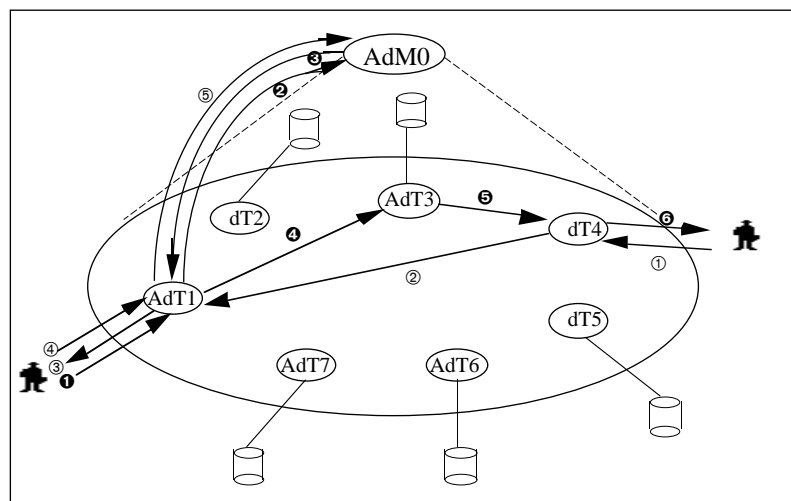


Figure 8.5.1. La relaxation restreinte (réalisée lors de la communication ⑤)

Contexte : le tableau ci-dessous résume les résultats obtenus suite à l'insertion de req1 (émise par AdT1) dans les deux réseaux de croyances possédés par chaque agent constituant ce prototype. Nous supposons que AdM0 possède un réseau de croyances sur autrui vide. Il en est de même pour tous les AdTs présents dans le système sauf pour AdT3 qui sait que AdT4 cherche à acquérir une golf.

	Réseau de croyances sur soi	Réseau de croyances sur autrui
AdM0	AdT3	∅
AdT1	∅	∅
AdT3	∅	AdT4

AdT4	AdT4	∅
------	------	---

Fonctionnement : AdT1 reçoit une requête en provenance de son adaptateur ❶. Ne connaissant aucun AdT compétent, il se trouve dans l'état bloqué et demande de l'aide à son AdM ❷.

AdM0 reçoit la requête de AdT1 ❷, l'interprète et se juge compétent : en effet, AdT3, présent sur son site, semble compétent pour aider AdT1. AdM0 transmet donc la croyance qu'il possède sur AdT3 à AdT1 ❸.

AdT1 reçoit la croyance de AdM0 concernant AdT3 ❸, l'insère dans son réseau de croyances sur autrui et interprète à nouveau la requête reçue de son adaptateur. Il juge AdT3 complémentaire.

AdT1 transmet la requête à AdT3 ❹ (message sans destinataire). Or, lorsque AdT3 interprète le message provenant de AdT1 il se trouve incompetent : la croyance fournie par AdM0 était erronée. Connaissant AdT4 susceptible d'être pertinent, AdT3 relaxe alors le message à AdT4 ❺.

AdT4 reçoit le message en provenance de AdT1 ❺ (mais qui en fait a été relaxé par AdT3). Il est bel et bien compétent et ne connaît personne de complémentaire. Il met à jour son réseau de croyances sur autrui et transmet la requête reçue à son adaptateur ❻. Ce dernier lui répond qu'il est d'accord pour acheter la voiture proposée par AdT1 ❼. AdT4 transmet alors la réponse de son adaptateur à l'agent demandeur, à savoir AdT1 ❷. AdT1 met à jour son réseau de croyances sur autrui et communique ensuite la réponse à son adaptateur ❸.

L'usager transmet son feed-back à AdT1 ❹ qui met à jour son réseau de croyances sur autrui et qui retransmet le feed-back à AdM0 ❺. AdM0 met alors à jour son réseau de croyances sur soi.

8.5.2. Mise en évidence de la communication spontanée

Contexte : le tableau ci-dessous résume les résultats obtenus suite à l'insertion de req1 (émise par AdT1) dans les deux réseaux de croyances possédés par chaque agent constituant ce prototype. Nous supposons que AdM0 possède un réseau de croyances sur autrui vide. Il en est de même pour tous les AdTs présents dans le système. La trace de cet exemple est fournie en annexe B.

	Réseau de croyances sur soi	Réseau de croyances sur autrui
AdM0	AdT4	∅
AdT1	∅	∅
AdT4	AdT4	∅

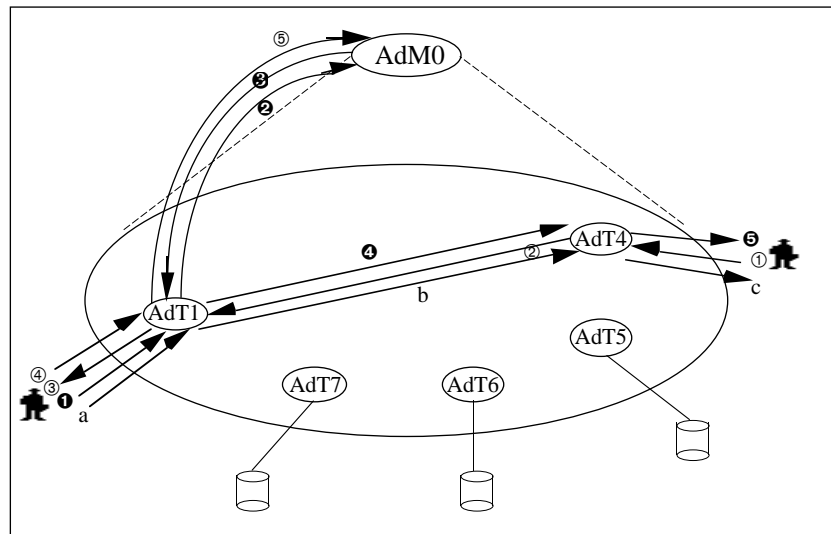


Figure 8.5.2. : La communication spontanée (réalisée lors de la communication b)

Fonctionnement : AdT1 reçoit une requête en provenance de son adaptateur ❶. Ne connaissant aucun AdT de compétent, il se trouve dans l'état bloqué et demande de l'aide à son AdM ❷.

AdM0 reçoit la requête de AdT1. Il l'interprète et se juge compétent car AdT4, situé chez lui, semble compétent pour aider AdT1. AdM0 transmet donc la croyance qu'il possède sur AdT4 à AdT1 ❸.

AdT1 reçoit la croyance de AdM0 concernant AdT4 ❹, l'insère dans son réseau de croyances sur autrui. Il interprète à nouveau la requête reçue de son adaptateur. AdT4 est jugé complémentaire de lui. AdT1 transmet alors la requête à AdT4 ❺ (message sans destinataire).

AdT4 reçoit le message envoyé par AdT1 ❻. Il se juge bel et bien pertinent et ne connaît personne de complémentaire. Il met à jour son réseau de croyances sur autrui et transmet la requête reçue à son adaptateur ❼. Ce dernier lui répond qu'il est d'accord pour acheter la voiture proposée par AdT1 ❶. AdT4 transmet alors la réponse de son adaptateur à l'agent demandeur, à savoir AdT1 ❷. AdT1 met à jour son réseau de croyances sur autrui puis transmet la réponse à son adaptateur ❸. Ce dernier lui retourne son appréciation ❹. AdT1 met à jour son réseau de croyances sur autrui et communique le feedback à son AdM pour qu'il puisse actualiser son réseau de croyances sur soi.

Quelque temps plus tard, l'utilisateur représenté par AdT1 prend connaissance d'une offre plus importante pour sa voiture. Il décide de faire savoir à AdT4 que sa voiture a déjà été vendue. L'utilisateur transmet la requête niée à son adaptateur (a) sans lui spécifier d'adresse logique car il a égaré l'adresse électronique d'AdT4. AdT1 reçoit donc une information non signée en provenance de son adaptateur (a). Il va alors la communiquer spontanément aux agents susceptibles d'être intéressés (b).

AdT4 reçoit le message transmis par AdT1 (b). Il se juge compétent et ne connaît personne de complémentaire ni de concurrent : il le transmet à son adaptateur (c).

Les diverses communications sont mises en évidence dans la figure 8.5.2.

8.5.3. Mise en évidence de la combinaison de services

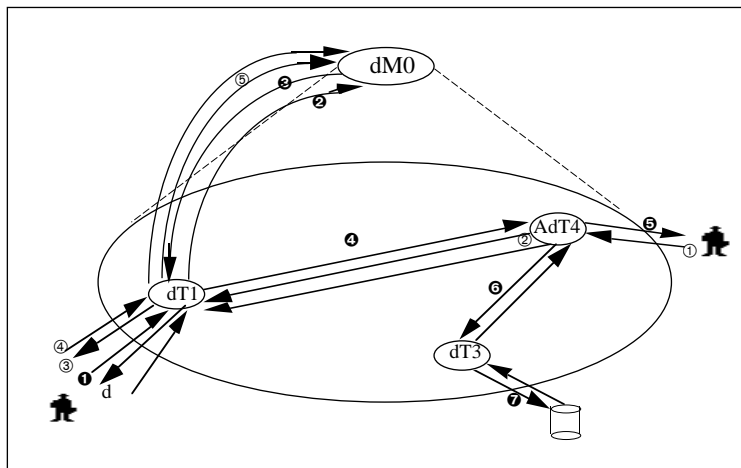


Figure 8.5.3. : La combinaison de services (réalisée lors des communication ⑤ et ⑥)

Contexte : le tableau ci-dessous résume les résultats obtenus suite à l'insertion de req1 (émise par AdT1) dans les deux réseaux de croyances possédés par chaque agent constituant ce prototype. Nous supposons que AdM0 possède un réseau de croyances sur autrui vide. Il en est de même pour tous les AdTs présents dans le système sauf pour AdT4 qui connaît AdT3.

	Réseau de croyances sur soi	Réseau de croyances sur autrui
AdM0	AdT4	∅
AdT1	∅	∅
AdT3	AdT3	∅
AdT4	AdT4	AdT3

Fonctionnement : il est identique à celui explicité en 8.5.2. jusqu'à ce que AdT4 reçoive la requête de AdT1 ④. AdT4 interprète cette dernière, se juge compétent, mais cette fois-ci, il connaît également AdT3 qui lui est complémentaire.

AdT4 communique la requête reçue à son adaptateur ⑤ et la transmet également à AdT3 en ayant au préalable mis son adresse comme expéditeur ⑥.

L'adaptateur de AdT4 lui répond ①, AdT4 transmet la réponse à l'agent demandeur ②, en l'occurrence AdT1.

AdT3 reçoit, quant à lui, la requête en provenance de AdT4 ③ et se juge compétent. Il met à jour son réseau de croyances sur autrui et transmet la requête à son adaptateur ⑦. L'adaptateur de AdT3

répond à AdT3 (a) qui retourne la réponse à l'agent demandeur en l'occurrence AdT4 (b). AdT4 met à jour son réseau de croyances sur autrui et retransmet la requête à l'agent demandeur (déterminé grâce à la table des objectifs), c'est-à-dire AdT1 (c).

AdT1 reçoit par conséquent deux réponses distinctes du même agent (② et c). Ces réponses sont fournies une à une à l'adaptateur de AdT1 (③ et d) qui fournira alors deux feed-back (④ et e). A la réception de chacun d'eux, AdT1 met à jour son réseau de croyances sur autrui, puis le communique à AdM0 (⑤ et f).

8.6. DISCUSSION

Le système multi-agent d'agents de transaction, représente le niveau intermédiaire de l'architecture d'ARCADIA. Il est constitué d'AdTs. Le but d'un AdT est d'être en situation coopérative avec son environnement constitué des AdTs situés sur le même site, de l'AdM auquel il appartient et de l'usager ou le service qu'il représente. Les AdTs interagissent entre eux en fonction des croyances qu'ils possèdent les uns sur les autres.

Tout comme au niveau des AdMs, il est possible de distinguer deux sortes d'apprentissage : celui concernant la mise à jour du réseau de croyances sur les compétences de l'agent de transaction suite au feed-back de l'usager (resp. service), et celui concernant la mise à jour du réseau de croyances sur autrui de l'AdT lors d'une réponse fournie par un autre AdT.

Les AdTs prennent effectivement part à la fonction globale du système sans qu'elle soit explicitement codée chez eux. En effet, ils permettent de mettre en relation un usager avec un ou plusieurs service(s) susceptible(s) de pouvoir lui permettre de faire avancer la résolution de la requête en cours. De plus, outre cette mise en correspondance de services et usagers compatibles, les AdTs permettent de faire également de la composition de services pour répondre de manière plus complète à la demande.

Lorsqu'un nouvel usager ou un nouveau fournisseur de services s'abonne à ARCADIA il est encapsulé avec les comportements coopératifs que nous préconisons. En aucun cas il peut avoir un comportement malhonnête visant à perturber le reste de la société. Si, toutefois un fournisseur de service voulait frauder en se disant plus compétent que ce qu'il est réellement, les croyances le concernant se réajusteraient au fur et à mesure des interactions effectives et des rétroactions négatives jusqu'à refléter la vraie image de ce dernier et non celle qu'il prétendait avoir initialement. Bien que le réseau puisse s'ajuster à ces perturbations, il faut préciser qu'elles ne peuvent survenir car la coopération est une attitude sociale rémanente dont sont dotés tous les AdTs par encapsulation dès leur création.

Nous venons de mettre en évidence, au travers de divers exemples, certains comportements coopératifs possédés par les AdTs. Ces comportements sont à la base de l'auto-organisation du système car ils entraînent des modifications de liens entre agents en fonction des interactions vécues. Ainsi si un agent doit résoudre une requête pour laquelle il n'est pas compétent il va bénéficier de l'aide d'agents pertinents. Cette demande peut entraîner, en particulier dans le cas d'une relaxation restreinte, de nouveaux liens d'interaction. Si peu de temps après, le même agent est sollicité pour résoudre une requête similaire, la situation non coopérative d'incompétence ne sera plus détectée. L'agent aura appris et mis à jour son réseau de croyances sur autrui : le temps de traitement s'en trouvera nettement amélioré.

Du point de vue des résultats, nous n'avons pas pu apporter d'éléments concrets de comparaison avec des outils comparables comme par exemple les moteurs de recherche d'information par manque de moyens techniques, matériels et humains. Une telle comparaison aurait demandé la mise en oeuvre d'un prototype à grande échelle composé d'un grand nombre de services et également d'un grand nombre d'adaptateurs. D'autre part, un travail d'une telle envergure aurait également nécessité une interrogation régulière et continue du système afin que celui-ci puisse apprendre et s'auto-organiser en conséquence.

Quoiqu'il en soit les propriétés de la partie 2 nous assurent qu'avec une méthode basée sur les principes d'auto-organisation dérivés (comme cela est fait dans ARCADIA), l'adéquation fonctionnelle du système est bien celle définie par son concepteur.

CHAPITRE 9

LE SYSTÈME MULTI-AGENT D'AGENTS DE CROYANCES

9.1. INTRODUCTION

Les AdMs et les AdTs ont une activité sous-jacente importante qui consiste à gérer dynamiquement leurs croyances sur autrui. Rappelons que :

- les AdMs ont des croyances sur les compétences d'autres AdMs,
- les AdMs ont des croyances sur les compétences des AdTs qu'ils contiennent,
- les AdTs ont des croyances sur les compétences d'AdTs situés dans le même AdM qu'eux,
- les AdTs ont des croyances sur leurs propres compétences.

Ces croyances sont employées soit pour rechercher des agents pertinents vers qui diffuser une information, soit pour interpréter au mieux les informations (peu ou pas structurées) qu'ils reçoivent. Elles sont aussi réactualisées après le résultat effectif d'une transaction entre AdMs ou AdTs.

ARCADIA est composé d'un grand nombre de services dont le contenu est très divers. Pour pouvoir construire un répertoire de services intéressants les agents ont donc besoin de gérer un grand nombre de croyances. Le réseau de croyances doit également prendre en compte les **différentes facettes d'un même service** donc les divers problèmes auxquels il peut répondre. D'autre part, étant donné la forte dynamique d'ARCADIA, un service peut être amené à disparaître momentanément rendant alors inexactes les croyances qui s'y rapportaient ; le réseau de croyances doit donc autoriser **la reconsidération de croyances exprimées en son sein**. Enfin, les croyances étant distribuées au sein de chaque agent et reflétant seulement le point de vue de l'agent auquel elles appartiennent, deux agents peuvent avoir une vue différente voire même contradictoire d'un même service. Aussi, il ne faut pas perdre de vue que les réseaux de croyances peuvent avoir des redondances ou des incohérences.

Les usagers, quant à eux, ont la possibilité de **formuler leurs besoins dans un format très libre**. Le réseau de croyances doit donc permettre de reformuler les requêtes le cas échéant. Il ne doit pas, en revanche, nécessiter de techniques très compliquées pour sa mise en oeuvre et ne doit pas en particulier demander de très gros efforts d'adaptation de la part des usagers. Enfin, au sein d'ARCADIA, des milliers de transactions s'effectuent en permanence ; il est donc impossible

d'avoir une connaissance globale du système à un instant donné. Les croyances sont finalement des connaissances sur le système qui peuvent être exploitées pour avoir une vue générale et approximative du système à un instant donné.

Pour toutes ces raisons, le choix a porté sur un système multi-agent auto-organisateur permettant de représenter les croyances possédées par les agents appartenant aux deux niveaux supérieurs de l'architecture proposée. L'utilisation de réseaux sémantiques aurait été une méthode plus simple et moins originale pour aborder le problème mais la dynamique et l'incertitude inhérente au système n'auraient pas pu être prises en compte avec un tel outil.

Le réseau s'auto-organise en fonction de la résolution des situations non coopératives et de la pertinence de l'agent par rapport à la requête. A ce niveau apparaît le troisième système multi-agent auto-organisateur d'ARCADIA. Il a les mêmes principes de fonctionnement que les deux autres.

Le rôle de chacun des réseaux de croyances est double :

- il doit permettre de **déterminer**, pour une requête donnée, l'**agent de même niveau connu** pertinent par rapport à la requête en cours de résolution (cf. § 9.3.1.)
- il doit également permettre la mise à jour des croyances de l'agent courant en fonction d'un feed-back (cf. § 9.3.2).

Il peut également permettre de construire une structuration de la requête qui permettra d'interroger le ou les service(s) élu(s), et éventuellement de déterminer les questions à poser à l'utilisateur pour compléter une requête. Nous n'avons pas exploité ces deux dernières possibilités.

Nous distinguons dans le système multi-agent de croyances quatre types d'agents que nous allons à présent décrire.

9.2. TYPOLOGIE DES AGENTS

Parmi les agents représentant les croyances, quatre types d'agents peuvent être distingués :

- l'agent **atomique** : qui contient un terme issu d'une requête déjà saisie. Un même terme est représenté par un unique agent dans le réseau.
- l'agent **requête** : qui est un agent regroupant tous les agents atomiques contenus dans une requête donnée. Autant d'agents requête que de requêtes reçues seront créés mais ces agents tendront à disparaître au fur et à mesure du fonctionnement du système.
- l'agent **service** qui est un agent atomique contenant l'étiquette d'un service réel particulier.
- l'agent **frontière** qui, comme son nom l'indique fait office de frontière avec l'extérieur du système. L'agent frontière est unique. C'est lui qui fait le lien entre l'intérieur du réseau et l'extérieur ; il reçoit les requêtes de l'utilisateur et active les agents atomiques contenus dans la requête émise. De même, c'est lui qui reçoit le feed-back du service interrogé et le réinjecte dans le réseau.

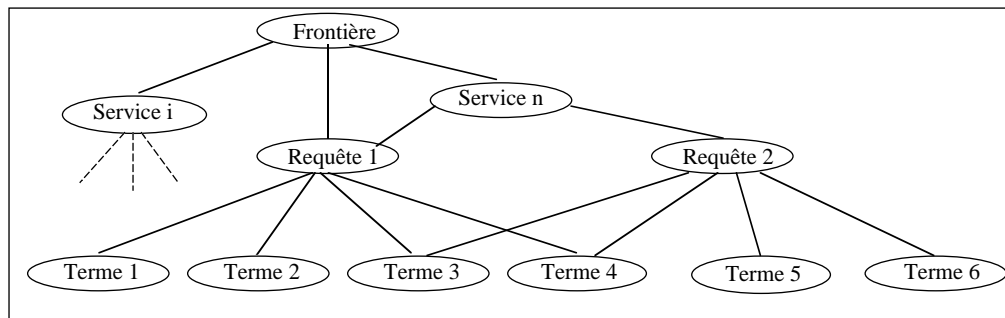


Figure 9.2. : Les quatre types d'agents de croyances

Tout agent atomique possède à la fois des agents fils et des agents pères :

- les **agents pères** d'un agent X sont les agents vers qui l'agent X émet de l'information lorsqu'il est activé,
- les **agents fils** d'un agent X sont les agents qui, une fois activés, lui envoient à leur tour un message d'activation.

On distingue également les **agents concurrents**. Ce sont des agents ayant les mêmes compétences que l'agent X. Ce dernier pourra leur déléguer des tâches en cas de surcharge de travail.

Ainsi, il est possible de dire que l'agent frontière est le père implicite de tous les agents services et le fils implicite de tous les agents termes.

9.3. FONCTIONNEMENT DU RÉSEAU D'ADC

Dès qu'un agent (AdM ou AdT) reçoit une requête, il insère cette dernière dans son réseau de croyances sur soi et dans son réseau de croyances sur autrui.

9.3.1. Insertion d'une requête dans le SMA

Supposons que la requête soit insérée dans le réseau de croyances sur autrui d'un AdT. La requête va parvenir, dans un premier temps, à l'agent frontière qui va activer les mots contenus dans la requête qui sont déjà présents dans le réseau. Chaque mot est unique dans le réseau donc représenté par un seul agent. Les agents vont alors propager le signal à leur père en suivant des règles de coopération que nous allons décrire au paragraphe 9.6.2. La propagation du "stimulus" est très proche de celle réalisée dans les réseaux de neurones. Lorsqu'un agent de croyance reçoit un message, il effectue une sommation des poids exprimant les croyances qu'il possède sur les

agents lui ayant fait part du message et compare cette somme à un seuil. Si la somme est supérieure à ce seuil il émet alors à son tour.

Lorsque le système se trouve dans un état coopératif, la propagation des signaux des agents de croyances contenus dans la requête va entraîner l'activation d'un ou plusieurs agent(s) service(s) connu(s) par l'agent courant : il (ils) sera (seront) dit(s) pertinent(s) pour la résolution de la requête courante. Le signal qui se propage n'est pas simplement un influx nerveux comme dans les réseaux de neurones mais contient les compétences des agents qui ont déjà été activés dans l'arborescence avant d'atteindre l'agent X. Ainsi, le message qui active l'agent Y étiqueté par le nom d'un agent représentant un service est déjà structuré. Il contient les termes qui ont permis de juger l'agent Y pertinent pour la requête saisie en entrée.

9.3.2. Le feed-back des transactions AdT/AdM

Une fois la transaction effectuée entre l'agent courant X et l'agent Y, la réponse est fournie à l'adaptateur de l'agent X. L'agent X va ensuite recevoir le compte-rendu de l'interaction en provenance de son adaptateur. Il va réinjecter le feed-back dans le réseau de croyances par le biais de son agent frontière. Cette insertion s'effectue en deux temps :

- L'agent frontière crée tout d'abord les mots contenus dans la requête qui n'existent pas déjà dans le réseau et active tous les mots contenus dans celle-ci. Un agent requête est également créé. Il est ensuite relié aux agents représentant les mots de la requête et à l'agent représentant le service. Ainsi pour une requête identique, l'accès au service compétent est direct.

- L'agent frontière va également envoyer un '!' vers le service élu ou un '~' suivant s'il a été pertinent ou pas.

Les actions effectuées par l'agent frontière vont entraîner des réorganisations dans le réseau. Celui-ci va alors s'adapter et apprendre, donc s'auto-organiser en fonction du feed-back reçu.

9.4. STRUCTURE DES AGENTS DE CROYANCES

La structure d'un agent de croyance est conforme à celle d'un AdT ou d'un AdM (cf. § 6.4.7.). Un AdC est composé :

- d'une **compétence** qui correspond au terme que l'agent représente,
- de **croyances** qui expriment la force d'un lien existant entre deux agents. Elles contiennent le rôle de l'agent sur lequel porte la croyance ainsi qu'une valeur (le poids) qui reflète l'importance accordée à l'autre agent,

- d'**aptitudes** à “raisonner” sur les messages qu’il a reçus de manière indépendante de la sémantique véhiculée par ces derniers. Il s'agit dans notre cas de la fonction générique que réalise chaque agent du réseau,
- d'un **langage d'interaction**. Les agents communiquent par envoi de messages. Un message est caractérisé par son émetteur (signature), son récepteur, la force illocutoire qui lui est associée (l'affirmation (“!”), l'interrogation (“?”), l'ignorance (“#”) et la négation (“~”)) et le contenu du message,
- d'une **attitude sociale** : la coopération.

9.5. COMMUNICATION CHEZ LES AGENTS DE CROYANCES

La communication entre agents de croyances nécessite, comme nous allons le voir, plusieurs étapes. De manière grossière elle peut être comparée au traitement d'un message par un neurone, mais uniquement pour la sommation des influx reçus et la comparaison à un seuil activateur.

9.5.1. La pondération des croyances

Chaque AdC possède des croyances sur les agents qu’il connaît. Un AdC communique avec un tiers grâce à la croyance qu’il détient sur ce dernier exprimée par l'intermédiaire d'un poids non nul (positif ou négatif). Il est positif lorsqu’il exprime une corrélation entre agents et négatif lorsqu’il exprime une opposition ou une incompatibilité. Un poids nul correspond à un agent oublié donc inconnu, celui-ci pouvant néanmoins continuer à émettre des messages s'il n'a pas lui-même un poids nul sur ce lien.

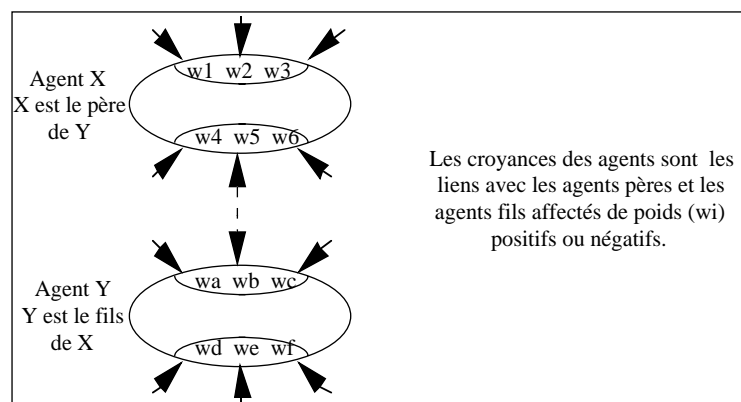


Figure 9.5.1. : Illustration des poids associés aux croyances

Considérons sur la figure 9.5.1. la croyance bidirectionnelle de l'agent X sur l'agent Y dessinée en pointillé. X a une croyance $w5$ sur Y et Y a une croyance wb sur X. Supposons que $w5 = 0$ et

$w_b > 0$. Ces hypothèses signifient que l'agent X ne connaît plus l'agent Y alors que l'agent Y le connaît. Ainsi l'agent Y peut continuer à envoyer des messages vers l'agent X. L'agent X recevra alors un message d'un agent qu'il ne connaît pas.

Les poids sur un lien ne sont donc pas réciproques. A l'initialisation, la valeur des poids associés aux croyances que possède un père sur ses fils est égale à : $(\text{Seuil} / \text{Nbre_fils}) + 1$.

La valeur des poids associés aux croyances évolue ensuite en fonction des situations non coopératives rencontrées au cours de l'histoire du couplage du système multi-agent de croyances et de son environnement.

9.5.2. Calcul de la somme des poids associés aux agents activateurs

Lorsqu'un agent reçoit un ou plusieurs message(s) en provenance de ses fils, il réalise la somme des poids des liaisons qu'il possède avec les fils qui lui ont envoyé une information de la manière suivante :

$$F = \sum_{i=1}^n (f_msg_i \times w_i)$$

avec f_msg_i désignant la force illocutoire du message i ('#' et '?' sont neutres), w_i le poids de la croyance envers l'émetteur du message i et n le nombre de messages

Contrairement aux '#' et '?' qui sont neutres, le '~' inverse le signe du poids w_i . Un poids positif devient donc négatif si la force illocutoire associée au message est un '~' tandis qu'un poids négatif associé à un message de type '~' devient positif.

9.5.3. La dynamique d'un agent

La somme calculée en fonction des messages reçus est comparée à un **seuil d'activation S** fixé. Si elle lui est supérieure, l'agent émet à son tour vers ses pères un message constitué de l'information reçue et de celle que lui-même détient. La sémantique de la fonction somme peut être un ET entre les activateurs (lorsque $\sum w_i = S$), ou un OU (lorsque $\forall w_i, w_i > S$).

Si la somme calculée est supérieure S, l'agent courant envoie un message à ses pères que nous nommerons dorénavant "**message d'activation**".

L'agent possède deux seuils d'activation : un positif et un négatif.

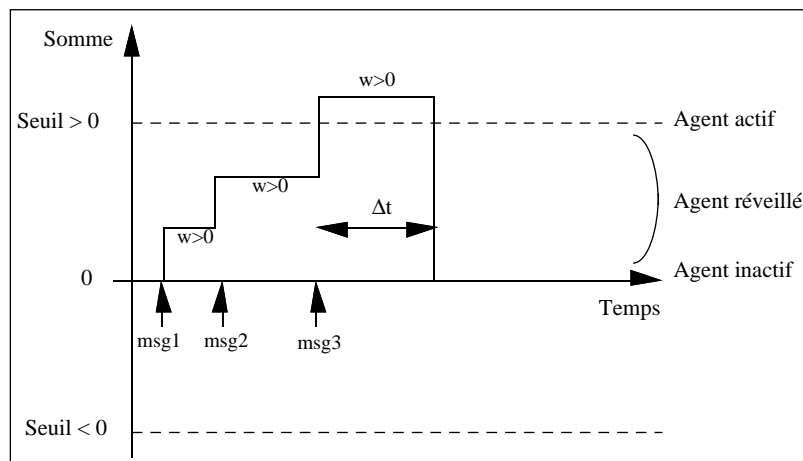


Figure 9.5.3. : Illustration du seuil d'activation d'un agent

9.5.4. Les trois états d'un agent

Quel que soit son rôle et suivant les interactions qu'il a eu, un agent peut être dans trois états. On dira qu'un agent est :

- **inactif** s'il n'a reçu aucun message d'activation de ses fils,
- **réveillé** s'il a reçu un ou plusieurs message(s) d'activation de ses fils mais si la somme des poids associés aux fils lui ayant envoyé un message d'activation, n'excède pas le seuil fixé. Un agent est donc réveillé si $\text{Seuil} > |F| > 0$,
- **actif** s'il a reçu un ou plusieurs messages d'activation de ses fils et la somme des poids associés à ces fils atteint le seuil. Un agent est donc actif si $|F| \geq \text{Seuil}$.

L'agent dans l'état actif communique avec son ou ses père(s) en lui (leur) envoyant un message d'activation.

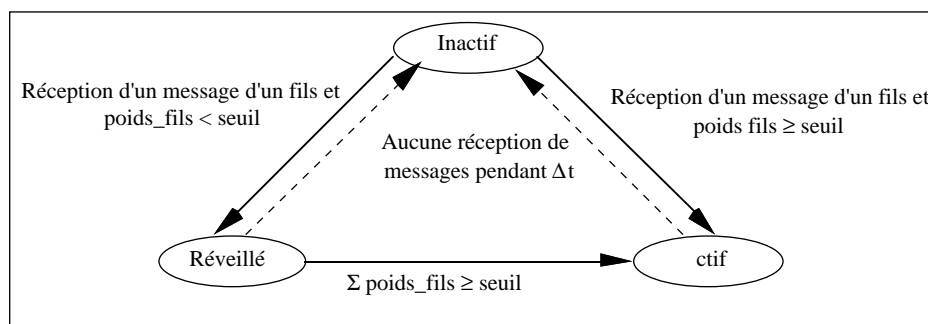


Figure 9.5.4. : Les changements d'états d'un agent

Un agent n'est actif que durant une période donnée (notée par la suite Δt) au delà de laquelle il redevient inactif. Plus précisément, lorsqu'un agent passe de l'état réveillé à l'état actif, il met en route un compteur initialisé à Δt diminuant au cours du temps. Tout message d'un fils lui parvenant

durant cet intervalle de temps est considéré et rajoute un Δt supplémentaire. Ainsi, la somme est calculée avec les poids des messages réceptionnés avant l'expiration du délai de temps imparti, ceci jusqu'à atteindre le seuil. Une fois le seuil atteint donc une fois l'agent actif, tout message reçu n'entraîne pas une mise à jour de la somme mais il est toutefois considéré pour détecter d'éventuelles situations non coopératives. Ce délai (Δt) est identique chez tous les agents et représente le temps acceptable entre deux messages corrélés (deux messages d'activation, question/réponse, ...). Il est utilisé pour faire repasser un agent à inactif entre deux requêtes.

9.5.5. Contenu des messages échangés entre agents

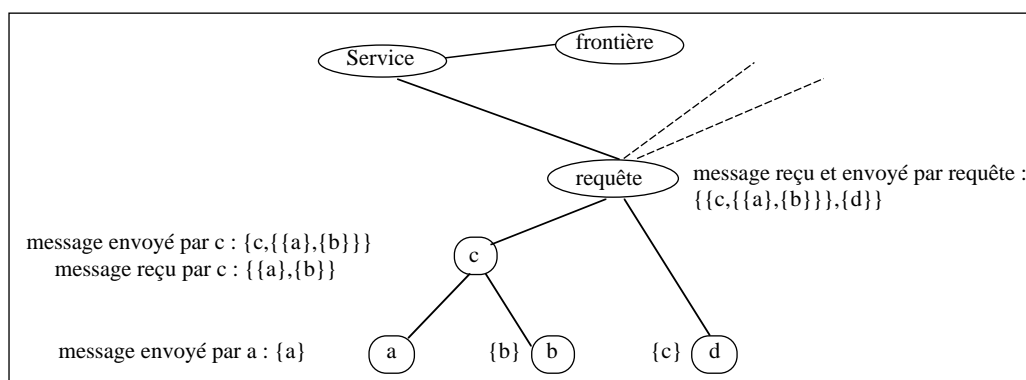


Figure 9.5.5. : Messages échangés entre agents

Lorsqu'un agent est actif, il envoie à son tour un message d'activation à son ou ses père(s). Il comprend une force illocutoire, l'adresse de l'émetteur, du récepteur ainsi qu'un contenu proprement dit. Ce contenu n'est autre que l'agrégation du contenu des messages d'activation que l'agent a reçu de ses fils, auquel il a rajouté son propre contenu.

9.5.6. La communication inter-AdCs

De manière générale, une croyance exprime la force et la nature (excitateur ou inhibiteur) d'un lien entre deux agents. A notre niveau, une croyance pour un agent représente le lien qui existe entre lui et l'agent concerné par la croyance ainsi que la force qu'il accorde à cette croyance. Elle lui permet également de connaître le rôle de l'agent pour lequel il possède une croyance : il saura en particulier s'il est pour lui un agent père, un agent fils ou un agent concurrent.

Les messages échangés entre AdCs reposent sur la terminologie du domaine. A la réception d'un message, l'agent fait la somme des poids des fils entrant dans la communication et compare cette somme à un seuil. Si elle lui est supérieure, l'agent communique à son tour à ses pères. Ce contenu n'est autre que l'agrégation du contenu des messages que l'agent a reçu de ses fils, auquel

il a rajouté son propre contenu. Il y a donc ainsi une propagation dans le réseau terminologique bien que les agents n'aient individuellement aucune connaissance des symboles qu'ils manipulent. La sémantique est totalement inexistante.

Lorsqu'un agent actif reçoit d'un de ses fils participant à la somme un message opposé à son activation deux comportements sont possibles :

- soit l'agent est encore actif malgré le message reçu (tendant à le faire passer dans l'état réveillé), auquel cas, il renvoie à ses pères le message précédemment transmis tel quel, et confirme son activité.

- soit l'agent passe dans l'état réveillé une fois le message reçu et considéré. L'agent va alors retourner le message précédemment envoyé à ses pères en changeant la force illocutoire et en la remplaçant par un '#', synonyme d'ignorance. Le renvoi d'un même message avec '#' comme force illocutoire montre la non monotonie des informations échangées entre agents. Du fait de l'asynchronisme, l'agent a reçu un signal qui a inhibé son activité donc qui remet en cause son ancienne activité. Il fait donc connaître ce changement à ses pères.

9.6. MISE EN OEUVRE DE L'INTERPRÉTATION CHEZ LES ADCS

La coopération idéale dans le cadre d'un réseau de croyances s'exprime de la manière suivante : la requête reçue par l'agent frontière est partiellement ou totalement contenue dans le réseau ; elle permet donc l'activation de termes dans le réseau qui permettent d'élire un ou plusieurs services pertinents pour la requête courante.

Un agent détectera pour lui-même une situation coopérative si un nombre suffisant de ses fils lui a permis de passer à l'état actif. Il enverra donc la totalité des informations qu'il possède vers ses pères. Un agent actif effectue aussi des activités complémentaires pour autrui comme des relaxations de messages d'agents inconnus de force illocutoire !, ~ ou ?, qui permettront à d'autres de créer de nouveaux liens. Si le scénario n'est pas identique à celui-ci, la coopération n'est pas totale, auquel cas, certaines situations non coopératives sont détectées localement.

De telles situations correspondent à un mauvais fonctionnement du réseau du point de vue de ou des agent(s) concerné(s) ; leur comportement vise alors à retrouver une situation coopérative par modification de leurs liens d'interaction c'est-à-dire de leurs croyances. La modification des poids des croyances, la création de croyance et la relaxation sont à la base de la dynamique du réseau.

Les agents de croyances détectent des situations non coopératives en fonction uniquement des types de messages échangés (force illocutoire et type de l'expéditeur) et non du contenu de ces messages. Le comportement générique attribué aux agents est donc totalement indépendant du domaine considéré.

9.6.1. Instanciation des situations non coopératives

Cinq des situations non coopératives génériques définies au paragraphe 5.2. se retrouvent dans le cadre du réseau de croyances. L'incompréhension et l'ambiguïté n'apparaissent pas : cela s'explique par l'absence d'interprétation du contenu des messages.

Il y aura :

- **incompétence** lorsque l'agent est réveillé mais loin de son seuil d'activation,
- **improductivité** lorsque l'agent est réveillé mais proche de son seuil d'activation,
- **inutilité** lorsque l'agent est actif mais ne sait pas à qui retransmettre le résultat issu de son raisonnement,
- **conflit** quand l'agent reçoit un message qu'il a émis mais nié de la part d'un récepteur,
- **concurrence** lorsque l'agent reçoit un message identique à celui qu'il possède.

9.6.2. Détection et traitement des situations non coopératives

Un agent détecte localement une situation non coopérative sans avoir besoin de considérer la "sémantique" véhiculée dans le message reçu car il réalise seulement des sommations de pondération et des appariements entre chaînes de caractères. Si une sémantique peut exister elle est simplement issue du raisonnement d'un observateur du système qui pourrait penser que les requêtes sont bien interprétées lorsque le réseau auto-organisateur qui s'est constitué est en interactions coopératives.

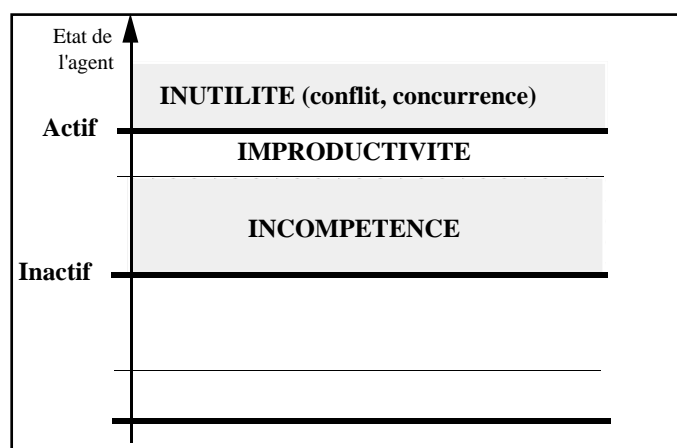


Figure 9.6.2. : Conditions des différentes situations non coopératives

9.6.2.1. L'incompétence

Un agent de croyance détecte une **incompétence**, lorsqu'il est réveillé par un certain nombre de fils mais de telle manière qu'il soit très loin du seuil d'activation.

Pour faire bénéficier autrui de l'activation de ses fils, il va les relaxer en direction de ses pères et de ses fils inactifs. Cela peut conduire à la création de nouvelles connexions dans le réseau.

9.6.2.2. L'improductivité

Un agent de croyance détecte une **improductivité** lorsque la réception d'un ou plusieurs messages lui permettent de passer de l'état inactif à réveillé mais il n'atteint jamais son seuil d'activation avant la fin de son Δt (mais il en est toutefois proche).

Lorsqu'une improductivité est détectée au Δt , il envoie un message avec une force illocutoire '?' vers ses fils inactifs et ses pères pour signifier une recherche de nouveaux fils. L'agent va tendre vers l'activation en ajustant les poids de ses fils actifs et aussi en ajoutant le cas échéant de nouveaux fils qui correspondent à des messages d'agents inconnus (car relaxés par ses voisins) et ayant une force illocutoire de type ! ou ~. Si dans le futur, une communication identique se reproduit, il pourra ainsi atteindre dans les mêmes conditions son seuil d'activation.

9.6.2.3. Le conflit

Un agent de croyance détecte un **conflit** avec son père lorsqu'il est actif (donc a précédemment émis un message à son père) et reçoit pourtant en provenance de ce dernier un message dont la force illocutoire est #.

Le traitement effectué par l'agent consiste à diminuer le poids du père et le poids des fils actifs, car ils sont tous responsables du conflit. Si l'agent est toujours actif après avoir baissé les poids de ses fils, il envoie à son tour un '#' à chacun de ses fils actifs : il y a alors propagation du conflit vers ses fils. Cette activité permet progressivement de couper des connexions entre agents si ce type de situation se produit régulièrement.

9.6.2.4. La concurrence

Un agent de croyance détecte une **concurrence** lorsqu'il reçoit d'un agent un message qui est identique à celui qu'il possède. Ainsi, deux agents sont concurrents s'ils ont les mêmes compétences : ils sont actifs simultanément et ils produisent les mêmes résultats.

L'agent qui vient de détecter la concurrence relaxe le message du concurrent vers le père puis ajuste ses poids. L'objectif de ce traitement de concurrence est soit la suppression ultérieure d'un

des deux agents (l'autre héritant des pères de l'agent supprimé), soit la spécialisation des deux agents par modification des liens avec leurs fils afin de ne plus produire les mêmes résultats.

9.6.2.5. L'inutilité

Un agent de croyance détecte une **inutilité** lorsqu'il passe dans l'état actif et ne possède pas de père à qui envoyer son message d'activation.

L'agent initie une relaxation vers ses fils inactifs dans le but d'atteindre un agent en état d'accepter un nouveau fils. Si cette situation perdure, l'agent disparaîtra.

9.7. RÉSULTATS OBTENUS A PARTIR DU SYSTÈME MULTI-AGENT D'ADCS

Dès l'intégration au sein d'ARCADIA d'un nouvel agent représentant un service ou un usager, il faut que le collectif déjà existant puisse en tirer immédiatement profit en effectuant des transactions pertinentes avec celui-ci. Cela impose un corpus de croyances déjà constitué relatif à ce nouvel agent, qui soit automatiquement constitué avec un fichier de requêtes types ; c'est un outil disponible dans ARCADIA. Les croyances sont bien entendu affinées et complétées pendant l'activité d'ARCADIA, à partir des résultats d'échanges ultérieurs avec cet agent. Les mesures qui sont données ici emploient ce mécanisme d'"apprentissage" initial.

Ce paragraphe présente les divers résultats que nous avons obtenus suite à l'implémentation du système multi-agent d'agents de croyances. Les résultats que nous allons exhiber dans ce chapitre ont été obtenus à partir du réseau de croyances sur soi du service pavehicules. L'apprentissage pour les croyances d'autres agents sont tout à fait similaires. Ce réseau de croyances pourrait correspondre aussi bien aux croyances sur soi de l'AdT pavehicules, qu'aux croyances sur soi de l'AdM qui le contient, qu'aux croyances sur autrui d'un agent qui aurait réalisé de nombreuses transactions avec pavehicules.

Détailler isolément chaque situation non coopérative ne permet pas de voir l'intérêt de leur contribution. C'est en effet la simultanéité de comportements différents entre agents voisins qui permet réellement l'auto-organisation. Nous allons donc nous intéresser au fonctionnement collectif du système et par là même montrer la pertinence des comportements coopératifs dont nous avons doté les AdCs au travers de divers exemples.

Le test du réseau de croyances sur autrui de pavehicules s'est effectué en plusieurs étapes :

- tout d'abord nous avons récupéré une à une les petites annonces postées sur les News dans le forum de discussion fr.petites-annonces.vehicules (284 au total) et les avons recopiées dans un fichier.

- ce fichier a été présenté en entrée de la méthode de construction automatique d'un réseau de croyances à laquelle nous avons demandé d'afficher, toutes les "x" requêtes, certaines caractéristiques afin de pouvoir les analyser et de nous rendre compte de l'apprentissage.

Le corpus de croyances initial a été acquis en injectant une cinquantaine de descriptions de requêtes réelles dans le réseau étudié. Nous allons à présent analyser les diverses caractéristiques fournies par le réseau durant sa constitution avec ces 55 requêtes. En situation réelle, cette apprentissage aurait pu se dérouler durant une longue période par une série de feed-back issus du résultat des transactions effectives entre services et usagers.

9.7.1. Proportion de messages non coopératifs

Un des problèmes sous-jacents à l'auto-organisation est sa conséquence sur les temps de réponse du système. Pour l'étudier nous avons compté les messages liés à des situations non coopératives (l'ordonnée du graphique) relativement au nombre total de messages d'apprentissage (l'abscisse) au cours de l'insertion des 55 requêtes. Les résultats obtenus sont visibles dans le schéma 9.7.1.

Nous pouvons donc remarquer que plus le nombre de messages d'apprentissage augmente dans le réseau plus le nombre de situations non coopératives croît. Cependant, nous pouvons noter que le nombre de situations non coopératives croît de manière beaucoup plus lente au fur et à mesure des apprentissages.

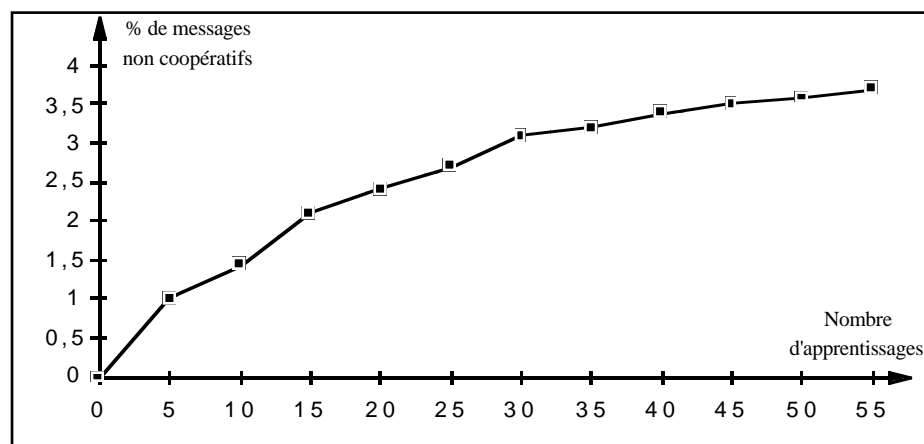


Figure 9.7.1. Proportion de messages non coopératifs au fur et à mesure de l'apprentissage

Ce résultat est conforme à ce que nous attendions puisque le but de l'apprentissage est de permettre au réseau de se stabiliser. Nous observons nettement un infléchissement de la courbe qui permet de penser que si nous avons un apprentissage effectué sur plusieurs centaines de requêtes, nous aurions une asymptote vers quelques pour-cent. Ces résultats nous permettent d'affirmer que si les messages de relaxation, les communications spontanées, les conflits sont indispensables à l'activité auto-organisatrice, ils grèvent peu le temps de réponse d'un réseau de croyances.

9.7.2. Étude d'agents particuliers

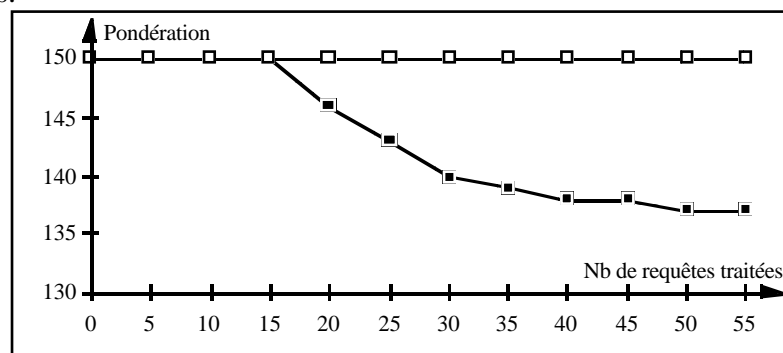
“pavehicles” est un service structuré en champs. Afin d'observer des comportements non coopératifs, nous avons volontairement laissé les noms des champs dans les requêtes d'apprentissage. Parmi ceux-ci, il est possible de distinguer le champ :

- from explicitant l'émetteur de l'annonce,
- subject décrivant le sujet traité dans l'annonce,
- date qui indique la date à laquelle l'annonce a été postée.

Ces trois termes apparaissant dans la quasi totalité des annonces, nous allons étudier leur évolution dans le réseau de croyances. Avant cela, nous allons nous intéresser à l'évolution du terme “pavehicles” qui est un agent particulier puisqu'il correspond au nom de l'agent détenteur du réseau de croyances étudié.

9.7.2.1. L'agent pavehicles

L'agent pavehicles possède autant de fils que de requêtes présentes dans le réseau. Ces requêtes sont initialisées avec un poids de 150 comme l'illustre le segment de droite supérieur du graphique ci-après.



Légende

—■— Pondération moyenne des agents "requête" présents dans le réseau

Figure 9.7.2.1. : Courbe des pondérations des agents “requêtes”

Comme plusieurs des requêtes insérées dans le réseau sont relativement voisines, nous pouvons nous attendre à ce que des activités concurrentes soient détectées dans le réseau. C'est effectivement ce qui se produit car la pondération illustrée par la seconde courbe diminue.

Cette diminution ne survient qu'après un certain nombre de requêtes car les premières sont suffisamment dissemblables pour ne pas être concurrentes au niveau de l'agent pavehicules.

Cette baisse est nécessairement bornée sinon cela signifierait que toutes les requêtes saisies seraient strictement équivalentes (on considère néanmoins que certaines requêtes vont réellement disparaître durant le fonctionnement du réseau).

9.7.2.2. Les agents "from", "subject", "date"

Parmi les 55 requêtes présentes dans le réseau, nous avons remarqué que 25% d'entre elles ne considèrent plus l'agent "from" comme leur fils (et réciproquement, l'agent from ne les considère plus comme ses pères non plus).

Autrement dit, le système multi-agent considère ce terme comme non signifiant par le seul traitement de situations non coopératives (principalement l'incompétence puisque dans un tel cas, l'agent relaxe ses fils actifs en direction de ses pères et de ses fils inactifs afin d'entraîner la création de nouvelles connexions dans le réseau et baisse les poids de ses fils actifs).

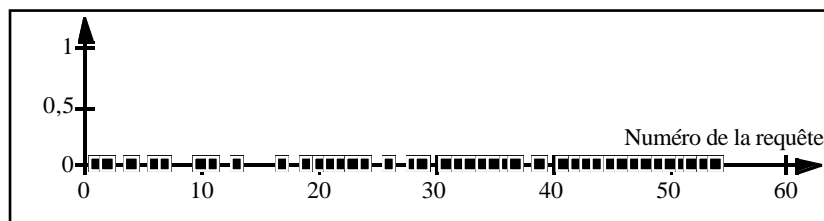


Figure 9.7.2.2. Évolution du terme "from" dans un réseau de croyances de pavehicules

Le graphique 9.7.2.2. représente, une fois le 55ème apprentissage effectué, les requêtes qui ont encore un fils "from". A chaque apprentissage, de nombreuses requêtes déjà intégrées dans le réseau diminuent progressivement le poids de leur fils "from". Cela justifie que la suppression des liens pères-fils ne soit pas uniforme mais beaucoup plus fréquente pour les premières requêtes car elles ont pris en compte beaucoup plus de situations d'incompétence que les dernières.

Le processus auto-organisateur a réalisé le même type de modification de croyances vis-à-vis des champs "subject" et "date".

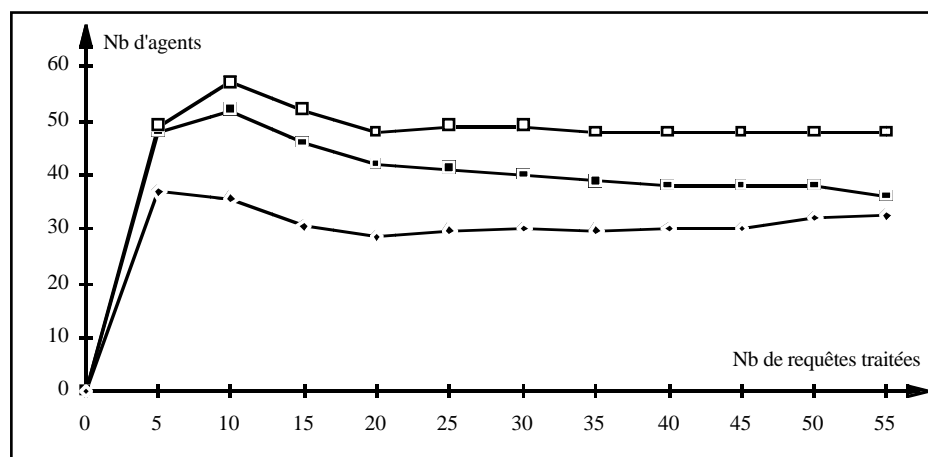
Ce processus d'élagage conduit effectivement à une représentation générique des compétences des services ou des usagers à partir de résultats de transactions spécifiques. Ce résultat simplifie systématiquement la connectivité mais ne supprime pas les informations principales dans les requêtes.

9.7.3. Nombre de termes par requête

Chaque offre d'usager peut contenir un nombre très variable de termes variant typiquement d'une vingtaine à cent vingt. Nous allons dans un premier temps étudier la proportion de termes actifs dans une requête puis dans un second temps les termes supprimés au fur et à mesure du processus d'auto-organisation.

9.7.3.1. Nombre de termes activés dans une requête

Les données précédentes sont relatives à la structure instantanée du réseau. L'activation effective d'un agent peut être réalisée avec un sous-ensemble restreint de ses fils, il suffit pour cela que la somme des poids des fils actifs dépasse son seuil d'activation (100). Si tous les fils d'une requête étaient nécessaires à son activation, cela correspondrait à un et logique. Sauf cas particuliers, cela devrait rarement survenir et nous devrions plutôt observer une certaine "redondance" entre les termes d'une requête.



Légende

- Nombre moyen d'agents par requêtes utilisateur
- Nombre moyen d'agents par requêtes présentes dans le réseau
- ◆— Nombre moyen d'agents actifs par requêtes actives

Figure 9.7.3.1. Nombre de fils actifs par requêtes actives / nombre moyens de termes

La seconde courbe du graphique 9.7.3.1. indique le nombre moyen d'agents par requête, une fois celles-ci insérées dans le réseau tandis que la troisième courbe montre le nombre moyen d'agents-fils actifs pour les requêtes actives durant la session. Les résultats sont conformes au présupposé précédent.

D'autre part, s'il y a une réelle activité auto-organisatrice, le réseau devrait supprimer des termes "non signifiants" et donc réduire la connectivité initiale. La première courbe reporte justement le nombre moyen de termes contenus dans les requêtes initiales des utilisateurs. Le résultat attendu est bien observé car la courbe 1 est située au dessus de la seconde (qui représente le nombre de termes moyens contenus dans les requêtes présentes dans le réseau). La réduction de la connectivité permet aussi de diminuer le nombre de messages envoyés et donc le temps de réponse.

9.7.3.2. Termes supprimés dans les requêtes

Pour juger si les termes supprimés sont ceux véritablement peu ou pas signifiants, nous avons repris ci-dessous une requête initiale dans laquelle les termes supprimés sont en gras. Rappelons que le processus auto-organisateur du réseau ne fait appel à aucune méthode d'analyse du langage naturel ni à une fonction d'évaluation globale.

from: ggourmel@ifremer.fr (Ghislaine Gourmelen,Ifremer DCB,98.22.40.07) **Subject:** **vends** BREAK **Date:** **29 Apr 1996** **Vends** BREAK VOLVO 240 GL (essence) - Turbo - **7** places **Année** : 1983 Etat exceptionnel (entretien Volvo) Toutes options **Prix** à débattre Téléphone : 98.47.73.73 après 18h (sinon répondeur)

Figure 9.7.3.2.a. : Analyse des termes supprimés dans une requête

Nous pouvons constater (9.7.3.2.a.) que 12 termes sur 48 au total sont supprimés. Parmi ceux-ci nous pouvons retrouver les termes "from", "subject" et "date" qui sont des termes usuels et visiblement sans signification pour activer la requête. Ce travail d'élagage qui ne fait appel à aucun traitement sémantique peut toutefois conduire à des modifications sémantiquement indésirables. Ces "erreurs" sont généralement corrigées par des apprentissages ultérieurs.

L'exemple ci-dessous illustre ces modifications indésirables : l'agent "pneus" est supprimé bien qu'il soit important dans la description de l'annonce. De même la suppression de l'agent "2" situé dans la requête à proximité du terme phase peut entraîner une ambiguïté au niveau de l'annonce : il est alors possible de lire *GT turbo phase 30* au lieu de *GT turbo phase 2*.

En revanche, nous constatons encore ici que les champs "from", "Subject" et "Date" sont supprimés.

from: ascensio@planete.net (Romain Soulié) **Subject:** Vds **5** jantes GT turbo phase **2** **Date:** **Tue, 30 Apr 1996** Je vends 5 jantes Gt turbo phase 2 **avec** factures **pneus** neufs michelin (2) **et** recam (2). **prix** : 2000 frs
Contacter moi **sur** ascensio@planete.net

Figure 9.7.3.2.b. : Analyse des termes supprimés dans une requête

9.7. DISCUSSION

Le système multi-agent de croyance, représente le niveau le plus bas de l'architecture d'ARCADIA. Il est constitué d'AdCs dont le seul but est d'être en situation coopérative avec les AdCs qu'il connaît (son environnement). Les AdCs interagissent en fonction des croyances qu'ils possèdent les uns sur les autres et en fonction d'un état interne qui les pousse ou non à interagir. Contrairement aux deux niveaux supérieurs, l'AdC modifie ses compétences et croyances dès lors qu'il interagit. Ces compétences sont modifiées en fonction des messages reçus qui tendent à le faire passer actif. Ces croyances sont modifiées lors des traitements des situations non coopératives.

On peut remarquer que bien que les AdCs poursuivent l'unique but d'être coopératifs avec les autres AdCs, ils participent efficacement à la gestion des croyances des AdTs et AdMs. Ils permettent, sans le savoir, de désigner un ou plusieurs service(s) pertinent(s) pour la requête reçue en entrée. Cette désignation ne repose que sur des considérations syntaxiques ne mêlant aucune considération sémantique. Il ne s'agit que de sommation de poids associés aux croyances, de manipulation de force illocutoire, de mise en oeuvre de la relaxation ...

Détailler isolément chaque situation non coopérative comme nous l'avons fait dans ce chapitre ne permet pas de voir l'intérêt de leur contribution. C'est en effet la simultanéité de comportements différents entre agents voisins qui permet réellement l'auto-organisation. La pertinence de tous ces comportements est mise en évidence dans les exemples donnés dans la quatrième partie.

Comme nous venons de l'observer sur cet exemple, l'activité auto-organisatrice est déjà significative avec une cinquantaine de messages d'apprentissage. Nous pourrions penser que des milliers de données soient indispensables à un système qui ne fait aucun présupposé sur le domaine à apprendre pour commencer à le structurer. Cela provient certainement du fait que nous avons programmé toutes les situations non coopératives issues de la théorie et pas seulement quelques méthodes qui auraient pu nous apparaître contextuellement pertinentes dans notre domaine particulier.

L'activité auto-organisatrice est simultanément fonction de la structure courante du réseau et de la perception d'un nouveau message par la frontière. Nous observons ainsi que l'histoire du couplage du système avec son environnement influe réellement sur la dynamique du réseau. En d'autres termes, l'ordre d'apprentissage des requêtes peut conduire à des structures différentes, mais qui portent principalement sur de faibles variations de pondérations pères-fils plus que sur l'organisation elle-même.

CONCLUSION ET PERSPECTIVES

1. DISCUSSION ET CONCLUSION

La complexité des applications à développer de nos jours fait qu'il est impossible d'aborder la conception des systèmes de manière usuelle, c'est-à-dire de manière descendante. Le concepteur de l'application se trouve tôt ou tard face à une situation qu'il n'avait pas prévue au moment des spécifications. Dès lors, une méthode de conception indépendante et suffisamment générique pour faire face à l'imprévu, devient un outil très précieux.

Ce travail de thèse propose une solution à cet effet. Elle est le fruit d'une constatation qui, somme toute, peut paraître évidente : il est vrai qu'intuitivement il semble naturel de penser que, lorsque tous les éléments d'un système entretiennent des relations coopératives les uns avec les autres, celui-ci fournit les résultats pour lesquels il a été conçu. Cette observation est l'objet de la réflexion menée dans la deuxième partie de ce manuscrit, réflexion qui nous a conduits à la démonstration de la propriété suivante : pour tout système fonctionnellement adéquat, il est possible de trouver au moins un système à milieu intérieur coopératif qui lui soit équivalent. Un système possède un milieu intérieur coopératif lorsque toutes les parties qui le constituent entretiennent des relations coopératives avec leur environnement et celui du système. Dès lors, le problème revient à savoir ce que veut dire "coopérer". La coopération doit à notre sens inclure au moins les aspects suivants : tout signal perçu par une partie constitutive du système doit pouvoir être interprétée sans ambiguïté, entraîner chez elle un certain raisonnement dont les conclusions doivent être utiles à son environnement. Un système respectant ces critères possède alors une organisation que nous avons qualifiée d'optimale. Cette théorie permet donc de garantir l'adéquation fonctionnelle d'un système mais aussi une méthode pour qu'un système s'auto-organise au mieux en fonction de l'histoire de son couplage avec l'environnement.

Dans un tel contexte, la non coopération survient lorsqu'au moins un des aspects précédents n'est pas vérifié. Les situations qualifiées de non coopératives découlant de cette considération sont alors au nombre de six. Ce sont les situations d'incompétence, d'improductivité, d'ambiguïté, d'inutilité, de concurrence et de conflit.

Afin de maintenir un état coopératif entre les éléments constitutifs d'un système, l'apparition d'une de ces situations doit entraîner un certain comportement visant à rétablir un état coopératif. Ainsi,

un élément du système doit être capable de détecter une de ces situations et transformer l'environnement dans lequel il évolue afin de revenir à une situation coopérative.

De telles considérations peuvent être interprétées comme une nouvelle manière de concevoir un système. Il ne devient alors plus nécessaire de connaître la finalité d'un système mais de savoir que les éléments qui le constituent doivent coopérer. Dès qu'une situation non coopérative survient, le système doit s'auto-organiser afin de revenir à une organisation optimale. La tâche d'un concepteur d'une application est alors double. Elle consiste à identifier et instancier les situations non coopératives dans le domaine dans lequel il se situe, puis à définir le traitement que doivent adopter les éléments du système lorsqu'ils y sont confrontés. Les parties du système s'auto-organisent ainsi en fonction des situations qu'elles entretiennent avec leur environnement.

Cette manière de concevoir un système a été appliquée à la problématique de la recherche d'informations dans le cadre d'un contexte distribué. Devant le succès grandissant d'Internet ainsi que la vitesse de prolifération des nouveaux services et des nouveaux abonnés, l'accès à un service particulier dans cet énorme réseau est un problème d'actualité et a fait l'objet d'un contrat CNET appelé ARCADIA. Sa finalité est de permettre l'accès à de l'information pertinente dans un réseau de sources d'information distribuées géographiquement, hétérogènes de par leur contenu et en constante évolution. ARCADIA est un système composé d'entités représentant respectivement un service ou un usager. Celles-ci tentent de s'adapter à leur environnement en fonction des interactions qu'elles ont entre elles.

Le fonctionnement d'un tel système est qualifié d'adéquat lorsqu'un usager satisfait désire se reconnecter au système et lorsque les services sont utilisés de manière la plus profitable pour le fournisseur de service. Conformément à notre propriété, afin de pouvoir atteindre un tel objectif, toutes les entités qui composent le système doivent être en coopération permanente les unes avec les autres.

Nous nous sommes alors intéressés dans un premier temps à définir et instancier les situations non coopératives pouvant survenir lors d'une recherche d'information dans un contexte réparti tout en essayant en parallèle d'entrevoir l'architecture la plus adaptée à la problématique. Après avoir défini trois types d'agents particulier, chacun correspondant à un niveau particulier de l'architecture, nous nous sommes intéressés à l'instanciation proprement dite des situations non coopératives. Cette étape a été réalisée en deux temps : elle a consisté d'abord à définir une méthode permettant de détecter chacune d'entre elles puis ensuite à définir les traitements devant être effectués par les agents. Ces deux étapes ont été définies dans l'optique d'être les plus indépendantes possible du domaine d'application et surtout d'une quelconque sémantique.

La méthode d'auto-organisation proposée repose sur des mécanismes basés sur des techniques multi-agents. Une de ces caractéristiques essentielles et non moins très intéressante est qu'elle accepte au départ l'impossibilité d'une vue exacte et centralisée du contenu du système ainsi qu'une spécification incomplète des situations pouvant survenir en cours de fonctionnement. L'erreur et le tâtonnement sont ainsi des activités courantes et ce sont leurs traitements qui permettent d'enrichir collectivement l'organisation. Ce processus d'auto-organisation est totalement transparent à l'utilisateur alors que les moteurs de recherche sont une couche d'accès supplémentaire et explicite.

Alors que les moteurs de recherche d'information existant sur Internet nécessitent une gestion d'index fastidieuse à mettre en oeuvre, cette fonctionnalité est automatique dans ARCADIA. Elle est réalisée implicitement dans ARCADIA : en effet, en interagissant avec les usagers et services, ARCADIA met à jour implicitement son organisation en s'auto-organisant de sorte à ce que les localisations soient réalisées de manière plus efficace. ARCADIA est également un système ouvert : l'ajout d'un usager ou d'un service se fait localement sur un AdM par création d'un nouvel AdT. Du point de vue des résultats, nous n'avons malheureusement pas pu apporter d'éléments concrets de comparaison car il aurait fallu tester notre application sur un domaine mettant en oeuvre un grand nombre de services. Cela était techniquement impossible. Cette impossibilité vient de plusieurs raisons :

- d'une part, la mise en oeuvre d'un prototype à grande échelle nécessite de pouvoir accéder à un grand nombre de services et donc de permettre un accès à chacun d'eux de manière rapide et efficace. Cette tâche, réalisée par l'équipe du CERT également impliquée dans le projet ARCADIA demande un grand investissement, en particulier lorsqu'on désire la rendre la plus générique possible.

- d'autre part, un fonctionnement à grande échelle nécessite une interrogation régulière et continue du système par des multitudes d'usagers afin que celui-ci puisse apprendre et s'auto-organiser en conséquence.

Cependant de manière intuitive, quoique cela ait été montré par diverses équipes (cf. chapitre 2 et annexe A), la coopération entre les agents permet à ces derniers de bénéficier de la connaissance des autres et donc dans le cas présent d'accéder de manière plus rapide, plus pertinente et optimisée à l'information comparativement aux moteurs de recherche.

D'autre part, le contrôle est implicitement encapsulé dans le code de l'agent. En effet, un agent cherche à être en permanence coopératif avec les autres agents et son comportement est implicitement dicté par cette recherche de coopération. Il ne nécessite par conséquent pas la mise en oeuvre de mécanismes particuliers supplémentaires.

Quoiqu'il en soit, le but de la théorie de la partie 2 est de garantir que le résultat fourni par le système est bien celui attendu par le concepteur de l'application. Bien que cela n'ait pu être vérifié

que sur un système constitué d'un nombre restreint d'entités, nous ne devons pas perdre de vue les résultats garantis par la propriété démontrée.

2. PERSPECTIVES

D'un point de vue plus particulier à ARCADIA, plusieurs perspectives sont envisageables outre le test à grande échelle de l'existant.

Une première problématique fort intéressante à la fois d'un point de vue recherche et d'un point de vue matériel (en particulier pour la réalisation de prototypes à grande échelle) serait de s'intéresser, lors d'un échec d'ARCADIA, donc lorsque la recherche d'un service particulier n'a pas abouti, de faire appel à un moteur de recherche d'information existant (Altavista, WebCrawler...). Étant donné l'envergure de tels moteurs (des millions de sites indexés) ils ont de grandes chances de fournir un résultat pour le problème posé et le travail consiste à permettre à un abonné d'ARCADIA de pouvoir accéder à un service localisé par le moteur.

La seconde problématique qui découle de la première serait de permettre à ARCADIA d'interroger les nouveaux services localisés par un moteur de recherche d'information comme il le fait actuellement via les adaptateurs. Cependant constituer des interfaces ad-hoc pour des millions de sites semble irréalisable. L'auto-organisation fondée sur la coopération pourrait là aussi être employée pour gérer les interfaces standards fournies par les sites. Elle pourrait reposer sur l'observation des interactions effectuées entre les usagers compétents et l'interface d'un service particulier. Le système ferait alors office de guide pour interroger un service particulier lorsque la personne est totalement novice en la matière.

Pour pouvoir valider une théorie scientifique, il faut réaliser de nombreuses expérimentations permettant de la confirmer ou de l'infirmer. Ainsi, l'idéal serait pour nous de pouvoir comparer les résultats obtenus par notre méthode à ceux obtenus en fonctionnement normal dans un système qui soit "observable". Un premier pas a été fait dans ce sens dans le projet du tileworld puis dans celui simulant le fourragement chez les fourmis (tous deux sont présentés très brièvement en annexe A). Les premiers résultats obtenus nous ont confortés dans notre idée de la coopération et nous ont encouragés à approfondir la théorie.

Nous espérons encore approfondir les résultats obtenus dans ces deux cadres d'étude ainsi que ceux acquis dans le projet ARCADIA. Nous souhaiterions également instancier notre méthode à d'autres domaines, voire même la mettre à disposition d'autres équipes afin qu'elles comparent les résultats obtenus avec leurs méthodes à ceux établis avec la nôtre.

Ceci est d'autant plus réalisable que notre méthode n'est pas destinée à résoudre un problème unique et particulier mais que c'est une méthode générale qui peut s'appliquer à plusieurs domaines de recherche.

BIBLIOGRAPHIE

[Anacleto, 1996]

Anacleto Louçã

“Points de vue sur l'apprentissage dans les systèmes multi-agents”

Journée Systèmes Multi-Agents du PRC-GDR IA, Toulouse - 2 février 1996. (pp. 47 - 57)

[Ashby, 1962]

Ashby W.Ross

“Principles of the self-organizing system”

in “Principles of self-organization” - H. Von FOERSTER, G. W. ZOPF Editors

Pergamon Press, 1962

[Bertalanffy, 1993]

Bertalanffy Von

“Théorie générale des systèmes”

Edition Dunod, 1993

[Bollen, 1996]

Bollen Johan, Heylighen Francis

“Algorithms for the self-organisation of distributed, multi-user networks. Possible application to the future World Wide Web”

Cybernetics and Systems'96 - R. Trappl (Ed.) - 1996

<http://pespmc1.vub.ac.be/papers/SelfOrgWWW.html>**[Bollen, 1995]**

Bollen Johan

“Adaptative Hypertext Networks That Learn The Common Semantics Of Their Users”

in Proceedings of the International Congress on Cybernetics

Namur - August 1995

<http://pespmc1.vub.ac.be/papers/namurart.html>**[Bonabeau, 1994]**

Bonabeau Eric, Theraulaz Guy

“Intelligence collective”

Editions Hermès - 1997 - ISBN : 2-86601-447-2

[Bonabeau, 1997]

Bonabeau Eric, Theraulaz Guy

“Auto-organisation et comportements collectifs : la modélisation des sociétés d'insectes”

dans “Auto-organisation et comportement” - Coordinateurs Guy Theraulaz et François Spitz -

Editions Hermès 1997 - p. 91 - 140

[Bouron, 1992]

Bouron Thierry

“Structure de communication et organisation pour la coopération dans un univers multi-agents”

Thèse de doctorat de l’université Paris 6 - Novembre 1992

[Bourret, 1991]

Bourret Paul, Reggia James

“Réseaux neuronaux : une approche connexionniste de l’intelligence artificielle”

Editions Teknea - 1991 - ISBN : 2-87717-016-0

[Brassac, 1996]

Brassac Christian, Pesty Sylvie

“‘La Pelouse Fourmilière’. De la coaction à la coopération”

Quatrièmes journées francophones sur l’Intelligence Artificielle Distribuée & Systèmes Multi-Agents, Port Camargue, 1-3 avril 1996, p. 251 - 264

Editions Hermès - 1996 - ISBN : 2-86601-528-2

[Camps, 1997b]

Camps V.

“ARCADIA : un système multi-agent auto-organisateur pour la recherche d’informations dynamiques et distribuées”

Rapport Interne IRIT/97-59R, Décembre 1997

[Camps, 1997a]

Camps V., Gleizes M.P., Trouilhet S.

“Properties analysis of a learning algorithm for adaptative systems”

First International Conference on Computing Anticipatory Systems

August 11-15 1997, Liège

[Camps, 1997b]

Camps V., Gleizes M.P.

“Une technique multi-agent pour rechercher des informations réparties”

Cinquièmes journées francophones sur l’Intelligence Artificielle Distribuée & les Systèmes Multi-Agents, 2 - 4 avril 1997, La Colle sur Loup

Éditions Hermès, pp. 29 - 46

[Camps, 1996a]

Camps V., Gleizes M.P.

“Cooperative and mobile agents to find relevant information in a distributed resources network”

Workshop on Artificial Intelligence-based tools to help W3 users

Fifth international conference on World Wide Web - 6 mai 1996 - Paris

<http://www.info.unicaen.fr/~serge/3wia/workshop/papers/paper30.html>

[Camps, 1996b]

Camps V., Gleizes M.P.

“Réflexions sur l'apprentissage en univers multi-agents”

Journée Systèmes Multi-Agents du PRC-GDR IA

Toulouse - 2 février 1996, pp. 59 - 68

[Camps, 1995]

Camps V., Gleizes M.P.

“Principes et évaluation d'une méthode d'auto-organisation”

Troisièmes journées francophones sur l’Intelligence Artificielle Distribuée & Systèmes Multi-Agents, St Baldoph, Savoie, 15-17 mars 1995, p. 337 - 348

http://www.irit.fr/ACTIVITES/EQ_SMI/PUBLI/VC95.html

[Camps, 1994]

Camps Valérie

“Une évaluation de la méthode de la relaxation restreinte pour l'auto-organisation dans les systèmes multi-agents”

Rapport de DEA “Représentation de la connaissance et formalisation du raisonnement” - IRIT - Toulouse - 1994

[Chevrier, 1993]

Chevrier Vincent

“Etude et mise en oeuvre du paradigme multi-agents : de Atome à Gtmas”

Thèse de l'université Nancy 1 - 11 juin 1993

[Conte, 1995]

Conte Rosaria, Castelfranchi Cristinano

Introduction of “Cognitive and Social Action”

p. 1-16 - UCL Press - ISBN : 1-85728-186-1

[Cesta, 1995]

Cesta Amedeo, Miceli Maria, Rizzo Paola

“Effects of Different Interaction Attitudes on a Multi-Agent System Performance”

Lecture Notes in Artificial Intelligence 1038 - “Agents Breaking Away”

Proceedings of the 7th European Workshop on Modeling Autonomous Agents in a Multi-Agent World, MAAMAW'96 - p. 128-138

Eindhoven, The Netherlands, January 1996 - ISBN : 3-540-60852-4

[CTI, 1997]

Rapport d'activité T0+24 du projet CNET “Accès intelligent à l'information répartie”

Bieber P., Camps V., Carpuat B., Cros P., Gleizes M.P., Glize P., Pezet J.

Contrat CTI CNET n°94CND46, janvier 1997

[CTI, 1996a]

Rapport d'activité T0+18 du projet CNET “Accès intelligent à l'information répartie”

Bieber P., Camps V., Carpuat B., Gleizes M.P., Glize P., Pezet J.

Contrat CTI CNET n°94CND46, juillet 1996

[CTI, 1996b]

Rapport d'activité T0+12 du projet CNET “Accès intelligent à l'information répartie”

Bieber P., Camps V., Carpuat B., Fraboul C., Gleizes M.P., Glize P., Pezet J.

Contrat CTI CNET n°94CND46, janvier 1996

[CTI, 1995]

Rapport d'activité T0+6 du projet CNET “Accès intelligent à l'information répartie”

Bieber P., Camps V., Carpuat B., Fraboul C., Gleizes M.P., Glize P., Pezet J.

Contrat CTI CNET n°94CND46, juillet 1995

[Davallo, 1989]

Davallo Eric, Naïm Patrick

“Des réseaux de neurones”

Edition Eyrolles - 1989 - p. 136-175

[Demazeau, 1990]

Demazeau Yves, Müller Jean-Pierre

“Decentralized Artificial Intelligence”

Proceedings of the First European Workshop on Modelling Autonomous Agents in a Multi-Agent World - Cambridge, England, August 16-18, 1989

Elsevier Science Publishers 1990 - ISBN : 0-444-88705-9

[Deneubourg, 1977]

Deneubourg J.L.

“Application de l’ordre par fluctuations à la description de certaines étapes de la construction du nid chez les termites”

Insectes Sociaux, n°24, p. 117 - 130

[de Rosnay, 1975]

de Rosnay Joël

“LE MACROSCOPE - Vers une vision globale”

Editions du seuil, 1975 - p. 94 - 97

[de Rosnay, 1997]

de Rosnay Joël

“Dictionary of Cybernetics”

6 janvier 1997

<http://pespmc1.vub.ac.be/FEEDBACK.html>

[Dowell, 1995]

Dowell Michael Lee

“Learning in Multi-Agent Systems”

PHD of Philosophy College of Engineering Department of Electrical and Computer Engineering
University of South Carolina - 1995

[Drogoul, 1993]

Drogoul Alexis

“De la Simulation Multi-Agents à La Résolution Collective de Problèmes : une étude de l’émergence de structures d’organisation dans les systèmes multi-agents”

Thèse de doctorat de l’université Paris VI - Novembre 1993

[Durand, 1979]

Durand Daniel

“La systématique”

Que sais-je? - Presses universitaires de France - 1979

[Ebstyne, 1996]

Ebstyne Michael J.

“The study of cooperative agents on dynamics problems”

April 15, 1996

<http://ulysses.stanford.edu/~michaele/thesis/Proposal.html>

[Etzioni, 1994]

Etzioni Oren, Weld Daniel

A Sofbot-Based Interface to the Internet.

Communications of the ACM - vol.37 - July 1994

[Ferber, 1995]

Ferber Jacques

“Les systèmes multi-agents : vers une intelligence collective”

InterEditions - 1995 - ISBN : 2-72960-572-x

[Ferber, 1994]

Ferber Jacques

“Coopération réactive et émergente”

Intellectica - 1994

[Foisel, 1997]

Foisel Rémy, Chevrier Vincent, Haton Jean-Paul
“Un modèle pour la réorganisation de système multi-agents”
Cinquièmes journées francophones sur l'Intelligence Artificielle Distribuée & les Systèmes Multi-Agents, 2 - 4 avril 1997, La Colle sur Loup
Éditions Hermès, pp. 261-277

[Foisel, 1996]

Foisel Rémy, Chevrier Vincent, Haton Jean-Paul
“De l'organisation d'une société à sa réorganisation”
Journée Systèmes Multi-Agents du PRC-GDR IA
Toulouse - 2 février 1996, pp. 121-128

[Forrest, 1990]

Forrest Stéphanie
“Emergent Computation : self-organizing, collective, and cooperative phenomena in natural and artificial computing networks”
Introduction to the proceedings of the ninth annual CNLS conference - p. 1-11 - 1990
Edited by Stéphanie Forrest - MIT Press - North Holland - ISBN 0-262-56057-7

[Gasser, 1991]

Gasser I., Ishida T.A
“Dynamic Organisational Architecture for Adaptative Problem Solving”
AAAI - 1991 - p. 185-190

[Genesereth, 1994]

Genesereth Michael R., Ketchpel Steven P.
“Software Agents”
Comm. of the ACM - vol.37 - 7 July 1994

[Glize, 1997]

Glize Pierre
“Do artificial systems need random for learning strategies ?”
First International Conference on Computing Anticipatory Systems
August 11-15 1997, Liège

[Glize, 1996]

Glize Pierre
“L'évolution du contrôle dans les systèmes multi-agents”
Actes du séminaire Programme REcherches Sciences COgnitives Toulouse sur “Le contrôle dans les systèmes coopératifs : le point de vue des sciences cognitives” - 9 et 10 mai 1996

[Goldman, 1994]

Goldman Claudia V., Rosenschein Jeffrey S.
“Emergent Coordination through the Use of Cooperative State-Changing Rules”
AAAI 1994 - p. 408 - 413

[Gurney, 1996]

Gurney Kevin
“Competition and self-organisation : Kohonen nets” - June 1996
<http://www.shef.ac.uk/psychology/gurney/notes/17/17.html>

[Guichard, 1996]

Guichard Frédéric
“La réorganisation dynamique dans les systèmes multi-agents”
Thèse de doctorat de l'Université de Savoie - 1996

[Haton, 1996]

Haton Jean-Paul
“Définition de mécanismes de coopération”
Bulletin de l'AFIA - n° 25 - avril 1996 - p.17

[Heurtevin, 1996]

Heurtevin Laurent
“Un système multi-agent auto-organisateur non cognitiviste pour la représentation de croyances”
Rapport de stage IUP “SI” - IRIT - Toulouse - 1996

[Heylighen, 1995]

Heylighen F.
“Self-organizing”
[CY] Web Dictionary of Cybernetics and Systems
<http://pespmc1.vub.ac.be/ASC/SELF-ORGANG.html>

[Hogg, 1993]

Hogg Tag, Huberman Bernardo A.
“Better Than The Best : The Power of Cooperation”
Lectures in Complex Systems - p. 163 - 184, Addison-Wesley 1993

[Holland, 1985]

Holland J.H.
“Properties of the bucket brigade algorithm”
Proc. International conference on genetic Algorithms and Their Applications -
Pittsburgh - p. 1-7

[IAD&SMA, 1996]

Actes des quatrièmees journées francophones sur l'Intelligence Artificielle Distribuée & Systèmes
Multi-Agents
Port Camargue, 1-3 avril 1996
Editions Hermès - 1996 - ISBN : 2-86601-528-2

[Jennings, 1994]

Jennings Nick
“Cooperation in industrial multi-agent systems”
World Scientific Series on Computer Science - Vol 33 -
World Scientific Publishing - 1994 - ISBN : 981-02-1652-1

[Kalenka, 1995]

Kalenka Susanne, Jennings Nick
“On Social Attitudes : A preliminary Report”
In DIMAS'95 - International Workshop on Decentralized Intelligent and Multi-Agent Systems
22 - 24 November 1995, Kracow, Poland - p. 233 - 240 - ISBN : 83-86813-10-5

[Kalenka, 1997]

Kalenka Suaanne, Jennings Nick
“Socially Responsible Decision Making by Autonomous Agents”
In Proc. Fifth Int. Colloq. on Cognitive Science (ICCS'97), Donostia, Spain, May 7-10, 1997

[Kirn, 1996]

Kirn Stephan
“Organizational Intelligence and Distributed Artificial Intelligence”
in Foundations of Distributed Artificial Intelligence
Edited by G.M. O'Hare and N.R. Jennings - 1996 - ISBN : 0-471-00675-0

[Krippendorff, 1997]

Krippendorff Klaus
"Dictionary of Cybernetics"
27 janvier 1997
<http://pespmc1.vub.ac.be/SELFORG.html>

[Lamp,90]

L. Lamport and N. Lynch
"Distributed Computing: Models and Methods"
in Handbook of theoretical computer science, Formal models and semantics, Elsevier and MIT Press, 1990, Editor Jan Van Leeuwen, Volume B, pp. 1157-1199.

[Lenay, 1996]

Lenay Christian
"Coopération et intentionnalité"
Quatrièmes journées francophones sur l'Intelligence Artificielle Distribuée & Systèmes Multi-Agents, Port Camargue, 1-3 avril 1996, p. 265 - 272
Editions Hermès - 1996 - ISBN : 2-86601-528-2

[Lesser, 1981]

Lesser V.R. & Erman L.D.
"Distributed Interpretation : A Model and Experiment"
Readings in DAI - 1981 - p. 120-139.

[Malone, 1988]

Maldone Thomas
"What is coordination theory"
In National Science Foundation Coordination Theory Workshop, MIT - 1988

[Marcenac, 1997a]

Marcenac Pierre, Calderoni Stéphane
"Self-Organisation in Agent-Based Simulation"
Proceedings of MAAMAW'97, Ronneby, Sweden, May 1997
<http://www.univ-reunion.fr/~marcenac/Publications/MAAMAW97.ps>

[Marcenac, 1997b]

Marcenac Pierre
"Modélisation de systèmes complexes par agents"
Revue de l'AF CET, Techniques et Sciences Informatiques, Editions Hermès, Septembre 1997
<http://www.univ-reunion.fr/~marcenac/Publications/TSI.ps>

[MARCIA, 1996]

MARCIA
"Auto-organisation : émergence de structures"
Journées du PRC GDR IA : les systèmes multi-agents, Toulouse, Février 1996

[Mataric, 1996]

Mataric Maja J.
"Learning in Muti-Robot Systems"
Lecture Notes in AI - n°1042 - 1996

[Mataric, 1994]

Mataric Maja J.
"Interaction and Intelligent Behavior"
PHD of Philosophy Massachusetts Institute of Technology May 1994

[Maturana, 1995]

Maturana Humberto R., Varela Francisco J.
“Overview of Autopoietic Theory”
<http://www.acm.org/sigois/auto/ATReview.html#observer>

[Maturana, 1992]

Maturana Humberto R., Varela Francisco J.
“L’arbre de la connaissance”
Édition Addison-Wesley - 1992 - ISBN : 2-87908-072-x

[Maturana, 1980]

Maturana Humberto R., Varela Francisco J.
“Autopoiesis and Cognition: The Realization of the Living”
Boston Studies in the Philosophy of Science
Cohen, Robert S., and Marx W. Wartofsky (ed.), Vol. 42, Dordrecht (Holland): D. Reidel
Publishing Co., 1980.

[Moraitis, 1994]

Moraitis P.
“Paradigme Multi-Agent et Prise de Décision Distribuée”
Thèse de Doctorat, Université Paris-Dauphine, 1994

[Morisset, 1997]

Morisset Benoît
“Etude d’algorithmes auto-organiseurs pour le fourragement de fourmis”
Rapport de DEA - IRIT/Paris Dauphine - Toulouse - 1997

[M. R. Jean, 1997]

Groupe de travail “collectif” IAD/SMA de AFCET/AFIA
“Emergence et SMA”
Actes des 5èmes journées francophones JFIADSMA’97 Intelligence Artificielle et Systèmes
Multi-Agents - Editions Hermès - 1997 - p. 323-342.

[Perez, 1988]

Perez Jean-Claude
“De nouvelles voies vers l’intelligence Artificielle”
Édition Masson - 1988 - ISBN : 2-225-81326-4

[Peyre, 1997]

Peyre Frédéric
“Technique de communication entre processus distants pour la recherche d’information répartie”
Rapport de stage de DUT Informatique Année Spéciale - juin 1997

[Piquemal, 1996a]

Piquemal-Baluard Christine, Carpuat Bernard
“Telemac : une application operationnelle d’une société d’agents implicitement coopératifs”
Quatrième journées francophones sur l’Intelligence Artificielle Distribuée & Systèmes Multi-
Agents, Port Camargue, 1-3 avril 1996, p. 31-33

[Piquemal, 1996b]

Piquemal-Baluard Christine, Glize Pierre
“Des aptitudes non cognitivistes d’agents pour l’auto-organisation”
Journée Systèmes Multi-Agents du PRC-GDR IA
Toulouse - 2 février 1996, pp. 129-138

[Piquemal, 1995]

Piquemal-Baluard C., Camps V., Gleizes M.P., Glize P.
“Cooperative agents to improve adaptivity of multi-agent systems”
Intelligent Agent Workshop of the British Computer Society
In Specialist Interest Group on Expert Systems & Representation and Reasoning
November 23th 1995, Oxford

[Piquemal, 1994]

Piquemal-Baluard Christine
“L'explication collective d'une société d'agents : conception d'un agent explicatif pour l'environnement Synergic”
Thèse IRIT UPS, 29 juin 1994

[Plaza, 1996]

Plaza Enric, Arcos Lluís, Martín Francisco
“Cooperative Case-Based Reasoning”
Distributed Artificial Intelligence Meets Machine Learning : learning in multi-agent environments - Lecture notes in artificial intelligence - Vol 1221 - ISBN : 3-540-62934-3
Gerhard Weiß Editor, Springer-Verlag, 1997 - p. 180-201

[Prasad, 1997]

Prasad Nagendra M V, Lesser Victor R and Lander Susan E
“Learning Organisational Roles for Negotiated Search in Multi-agent System”
Special Issue on Evolution and Learning in Multi-Agent Systems
In the International Journal of Human-Computer Studies (IJHCS) - 1997
<ftp://dis.cs.umass.edu/pub/or-ijhcs-naghi.ps>

[Prasad, 1996]

Prasad Nagendra M V, Lander Susan E and Lesser Victor R
“Cooperative Learning over Composite Search Spaces : Experiences with a Multi-agent Design System”
AAAI - 1996 - p. 68-73
<ftp://dis.cs.umass.edu/pub/naghi-cl-aaai.ps>

[Prigogine, 1994]

Prigogine Ilya
“Les lois du chaos”
Flammarion, 1994

[Renders, 1995]

Renders Jean-Michel
“Algorithmes génétiques et réseaux de neurones”
Editions Hermès - 1995 - ISBN : 2-86601-467-7

[Salles, 1997]

Salles Nicolas
“Plate-forme graphique de simulation d'insectes sociaux : les fourmis fourrageuses”
Rapport de stage de DUT Informatique Année Spéciale - juin 1997

[Searle, 1979]

Searle J. R.
“A taxonomy of illocutionary acts”
Expression and Meaning, Studies in the Theory of Speech Acts
Cambridge University Press 1979 - Chapitre 1 - p. 1-29.

[Sekaran, 1995]

Sekaran Mahendra, Sen Sandip
"To help or not to help"
Seventeenth Annual Cognitive Sciences Conference
July 22-25 - 1995 - Pittsburg Pennsylvania

[Sekaran, 1994]

Sekaran Mahendra, Sen Sandip
"Multi-Agent Learning In Non Cooperative Domains"
ECAI 94

[Sen, 1996a]

Sen Sandip
"Reciprocity : a foundational principle for promoting cooperative behavior among self-interested agents"
in Proc. of the Second International Conference on Multi-Agent Systems - p. 322-329 - AAAI Press - Menlo Park, CA, 1996
<http://euler.mcs.utulsa.edu/~sandip/icmas96-reciprocity.ps>

[Sen, 1996b]

Sen Sandip
Introduction au AAAI Spring Symposium Series 1996 : "Adaptation, Co-evolution and Learning in Multi Agent Systems"
Mars 1996
<http://euler.mcs.utulsa.edu/~sandip/ss.html>

[Sen, 1995]

Sen Sandip, Sekaran Mahendra
"Multi-agent coordination with learning classifieur systems"
Workshop on Adaptation and Learning in Multi-Agent Systems - IJCAI 1995 - August 20-25 1995, Montreal, Canada

[Sen, 1994]

Sen Sandip, Sekaran Mahendra, and Hale Jones
"Learning to coordinate without sharing information"
Proceedings of the twelfth national conference on Artificial Intelligence - Seattle Washington - July 1994
<http://euler.mcs.utulsa.edu/~sandip/aaai94.ps>

[Shaw, 1989]

Shaw M. J. and Whinston A.B.
"Learning And Adaptation In DAI"
in Gasser and Huhns - 1989 - p. 413-427.

[Sian, 1991]

Sian Sati Singh
"Adaptation Based On Cooperative Learning In MAS"
in Proceedings of the second workshop on Modelling Autonomous Agents in Multi-Agent World - Editors Y. Demazeau & J-P Müller - Editions North Holland - 1991

[SinghS, 1996]

Singh Satinder, Norvig Peter, Cohn David
"How to Make Software Agents do the Right Thing : An Introduction to Reinforcement Learning"
Adaptative system group - July 1996 - Harlequin Inc.
<ftp://ftp.cs.colorado.edu/users/baveja/Papers/agents.ps>

[SinghM, 1997]

Singh Munindar P., Huhns Michael N.

“Challenges for Machine Learning in Cooperative Information Systems”

Distributed Artificial Intelligence Meets Machine Learning : learning in multi-agent environments - Lecture notes in artificial intelligence - Vol 1221 - ISBN : 3-540-62934-3

Gerhard Weiß Editor, Springer-Verlag, 1997 - p. 11-24

[Smith, 1981]

Smith R.G. and Davis R.

“Frameworks for Cooperation in Distributed Problem Solving”

IEEE Transactions on Systems, Man, and Cybernetics - Vol. SMC-11 n°1 - January 1981 - p. 61-70

[Steels, 1996b]

Steels Luc

“The spontaneous Self-organization of an Adaptive Language”

Machine Intelligence 15 - Oxford University Press - Oxford - Muggleton S. (Ed.) - March 29, 1996

<http://arti.vub.ac.be/www/steels/mi15.ps>

[Steels, 1996a]

Steels Luc

“Self-organizing Vocabularies”

Proceedings of Alive V, Nara Japan - In Langton C. (Ed.) - 1996

<http://arti.vub.ac.be/www/steels/alife96.ps>

[Steels, 1990]

Steels Luc

“Cooperation between distributed agents through self organization”

Decentralized Artificial Intelligence - Vol 1 - Ed. Demazeau & Müller - Elsevier North Holland - 1990 - ISBN : 0-444-88705-9

[Stengers, 1983]

Isabelle Stengers

“Des tortues jusqu'en bas...”

Colloque de Cerisy - l'auto-organisation de la physique au politique

Paul Dumouchel, Jean-Pierre Dupuy - Edition du Seuil - Avril 1983

[Terabe, 1997]

Terabe Masahiro, Washio Takashi, Katai Osamu, Sawaragi Tetsuo

“A study of organizational learning in Multi-agent systems”

Distributed Artificial Intelligence Meets Machine Learning : learning in multi-agent environments - Lecture notes in artificial intelligence - Vol 1221 - ISBN : 3-540-62934-3

Gerhard Weiß Editor, Springer-Verlag, 1997 - p. 167-179

[Théraulaz, 1997]

Théraulaz Guy

“Auto-organisation et comportements collectifs dans les sociétés animales”

dans “Auto-organisation et comportement” - Coordinateurs Guy Théraulaz et François Spitz - Hermès 1997 - p. 79-83.

[Trouilhet, 1993]

Trouilhet Sylvie

“Représentation et traitement des connaissances sociales chez l'agent : application à l'environnement multi-agent SYNERGIC”

Thèse IRIT UPS, N° d'ordre : 1512, Toulouse, juillet 1993

[Ünsal, 1993]

Ünsal Cem

“Self-organization in large populations of mobile robots”

Master of Sciences in Electrical Engineering, May 1993, Blacksburg, Virginia.

<http://armyant.ee.vt.edu/unsalWWW/cemthesis.html>

[Varela, 1993]

Varela Francisco J., Thompson Evan, Rosch Eleanor

“L'inscription corporelle de l'esprit”

Editions du seuil - Avril 1993

[Varela, 1988]

Varela Francisco J.

“Autonomie et connaissance : essai sur le vivant”

Editions du seuil - 1988

[Von Bertalanffy, 1993]

Von Bertalanffy Ludwig

“Théorie générale des systèmes”

Editions Dunod - 1993

[Watkins, 1992]

Watkins C.J.C.H, Dayan P.

“Q-learning”

in Machine Learning - vol8 - p. 279-292.

[Weiß, 1993]

Weiß Gerhard

“Learning To Coordinate Actions In Multi-Agent Systems”

in Proceedings of the International Joint Conference on Artificial Intelligence - August 1993 -

p. 311-316.

[Weiß, 1996]

Weiß Gerhard

“Reader's Guide”

in Distributed Artificial Intelligence Meets Machine Learning : learning in multi-agent environments - Lecture notes in artificial intelligence - Vol 1221 - ISBN : 3-540-62934-3

Gerhard Weiß Editor, Springer-Verlag, 1997 - p. 1-10

[Wiederhold, 1992]

Wiederhold, Gio

“Mediators in the architecture of future information systems”

IEEE Computer Vol25 N°3, 1992 - pp38-49

[Wooldridge, 1995]

Wooldridge Michael, Jennings Nicholas R.

“A theory of cooperative problem solving”

Intelligent Agent Workshop of the British Computer Society

In Specialist Interest Group on Expert Systems & Representation and Reasoning

November 23th 1995, Oxford - p. 15-26

ANNEXE A

EXEMPLES DE SYSTÈMES REPOSANT SUR NOTRE MÉTHODE D'APPRENTISSAGE

A.1. INTRODUCTION

Nous allons présenter dans cette annexe une instanciation des situations non coopératives définies au chapitre 5 dans le cadre de deux exemples. Le premier est un jeu appelé "tileworld" [Piquemal, 1995], [Camps, 1996] le second concerne le fourragement chez les fourmis [Morisset, 1997].

A.2. APPLICATION AU JEU DU TILEWORLD

Nous avons appliqué la théorie proposée à un jeu, appelé "tileworld". Dans ce jeu, les agents se déplacent verticalement ou horizontalement sur un damier 10*10 sur lequel se trouvent des pavés, des trous et des barrières. Le but du jeu est de combler si possible tous les trous à l'aide de pavés. L'objectif premier de chaque agent est de pousser le pavé le plus proche dans le trou le plus proche. La notion de proximité n'est autre qu'un calcul de distance euclidienne qui ne tient pas compte des barrières, des autres agents et des pavés. Les agents se déplacent ensuite en contournant les barrières. Tous les agents possèdent les mêmes compétences et ont des croyances sur les autres : ils croient que les autres ont les mêmes compétences qu'eux.

L'interaction s'effectue par des perceptions et des actions sur le monde. Chaque agent observe le résultat du raisonnement individuel des autres agents à partir de ses croyances avant de se déplacer. Il présuppose ainsi le comportement des autres agents. La transmission de messages entre agents étant impossible, les agents ne peuvent donc ni négocier lorsque des conflits surviennent, ni informer les voisins en cas de concurrence ou d'ambiguïté.

Un agent perçoit tous les événements qui peuvent survenir sur la grille. Il connaît par conséquent l'existence de tous les autres agents. Il va donc améliorer son comportement en tenant compte de celui des autres.

Tout comme chez Goldman, qui a également étudié ce jeu [Goldman, 1994], notre simulation met en oeuvre des agents ayant une stratégie identique. Le fonctionnement de base d'un agent est

le suivant : il se dirige vers le pavé le plus proche, il pousse alors ce palet vers le trou le plus proche.

La durée d'une session est mesurée en nombre de cycles. Un cycle est l'unité minimale de temps pendant laquelle plusieurs agents peuvent se déplacer, en parallèle, d'une case à une autre. Tous les agents ayant la même vitesse de mouvement, l'application considère des agents synchrones : à chaque cycle, chaque agent effectue un déplacement élémentaire (s'il est possible compte tenu des contraintes de l'environnement). La session est terminée lorsqu'il ne reste plus de pavé ou de trou, ou lorsque le temps alloué à la session est terminé. Le nombre total de déplacements effectués ainsi que le nombre de cycles sont calculés au fur et à mesure de la session.

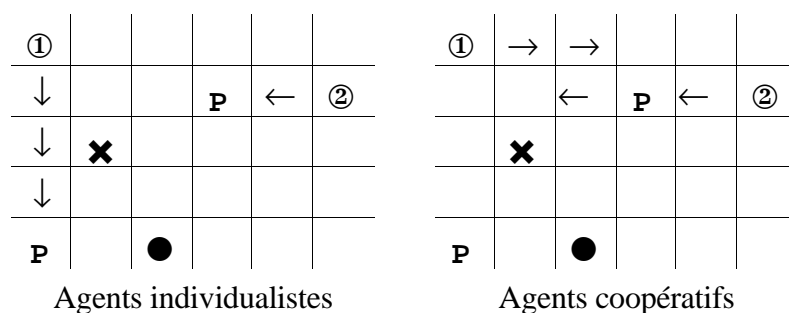
A.2.1. Détection et traitement des situations non coopératives

Dans les paragraphes suivants, nous détaillons trois des situations non coopératives définies en 5.2. : le conflit, l'ambiguïté et la concurrence. Nous ne pouvons pas mettre en évidence des comportements dans les cas de compétence et d'improductivité pour le domaine d'application choisi car tous les agents connaissent parfaitement le monde. L'inutilité est également inexistante car chaque action (déplacement) effectuée dans le monde est utile à autrui pour déterminer l'action à entreprendre.

Pour chaque situation non coopérative, nous montrons comment un agent la détecte puis nous présentons la manière de l'instancier dans le domaine d'application du "tileworld".

A.2.1.1. La situation non coopérative d'ambiguïté

L'ambiguïté dans le "tileworld" peut être formulée par : un agent est à la même distance de deux pavés ou de deux trous.



Légende : P pavé ● trou → sens de déplacement X barrière ① agent n°1

Figure A.2.1.1. : Comportement d'un agent dans le cas d'une situation ambiguë

Dans la figure A.2.1.1., l'agent A^1 a deux pavés qui sont à la même distance. Il détecte donc une ambiguïté. La grille de gauche indique le comportement d'agents individualistes qui ne se soucient pas des situations ambiguës. Comme ils ont des stratégies d'actions identiques, ils vont choisir arbitrairement de se diriger vers le pavé le plus proche. Ils sont alors amenés à pousser deux pavés dans le même trou ce qui provoque un conflit de trou.

La grille de droite montre le comportement d'agents coopératifs dans la même situation. A^1 va résoudre son ambiguïté en choisissant le pavé partagé avec A^2 pour coopérer avec ce dernier et travailler plus efficacement. L'agent A^2 va alors pousser le pavé pour l'amener dans une position telle que l'agent A^1 n'ait plus qu'à le pousser directement vers le trou.

La résolution de l'ambiguïté consiste donc à choisir avant tout les pavés convoités, c'est-à-dire ceux qui constituent les objectifs des autres, ceci afin de coopérer.

A.2.1.2. La situation non coopérative de concurrence

La concurrence dans le "tileworld" se traduit par : deux agents sont à la même distance de deux pavés ou de deux trous.

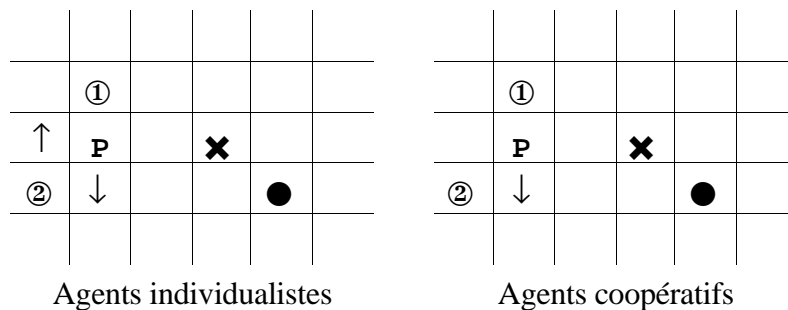


Figure A.2.1.2. : Résolution d'une situation de concurrence.

Dans la figure A.2.1.2., les agents A^1 et A^2 sont en concurrence : ils veulent pousser le même pavé dans le même trou.

Lorsque les agents sont individualistes, ils vont choisir de se diriger vers le pavé le plus proche d'eux, en l'occurrence ici l'unique pavé. Ils vont ainsi être amenés à vouloir pousser le même pavé dans le même trou. Un agent travaille inutilement puisque seul un pavé sera poussé dans le trou.

Lorsque les agents sont coopératifs et sont tous deux en situation de concurrence, ils vont tenter de s'aider réciproquement en coopérant. Pour cela A^1 va amener le pavé dans une position telle que A^2 n'ait plus qu'à le pousser directement vers le trou.

La résolution de la concurrence consiste à se coordonner pour atteindre le but plus efficacement.

A.2.1.3. La situation non coopérative de conflit

On distingue trois types de conflits dans le domaine du "tileworld".

- 1- Les conflits de trou : deux agents désirent amener deux pavés différents dans un même trou.
- 2- Les conflits de case : deux agents prévoient de se déplacer sur la même case.
- 3- Les conflits de case et de trou : deux agents prévoient de se déplacer sur la même case afin d'amener deux pavés différents dans un même trou.

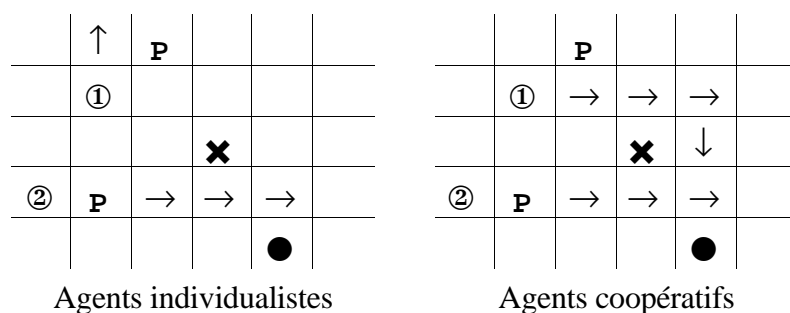


Figure A.2.1.3. : Résolution d'une situation de conflit

Dans la figure A.2.1.3., les agents A^1 et A^2 ont un conflit de trou parce que chacun désire le combler avec un pavé différent.

Lorsque les agents sont individualistes et ont des stratégies d'action identiques, ils vont choisir de se diriger vers le pavé le plus proche d'eux et d'amener ce dernier dans le trou le plus près. Ils vont ainsi être amenés à vouloir pousser leur pavé respectif dans le même trou. Ils vont être alors amenés à faire de nombreux déplacements inutiles et parfois même à se gêner mutuellement.

Lorsque les agents sont coopératifs et tous deux en situation de conflit, ils vont tenter de s'aider réciproquement. Pour cela A^1 et A^2 vont tout d'abord essayer de changer de trou : impossible car, dans le cas présent, le trou est unique. Ils essaient alors de changer de pavé. A^2 ne peut pas. A^1 , lui, détecte une ambiguïté. Comme dans le cas A.2.1.1., A^1 va choisir le pavé convoité par A^2 . Il choisit donc parmi les plans qu'il peut appliquer, celui qui le coordonne avec l'agent A^2 .

La résolution de conflit se résume à deux actions : éviter le conflit, ce qui signifie changer d'objectif (trou, pavé ou chemin) ou ignorer le conflit ce qui signifie laisser l'agent le plus rapide exécuter son plan.

A.2.2. Résultats expérimentaux et analyse

Afin d'étudier les avantages de notre approche, nous avons calculé le nombre d'actions réalisées par les agents en tenant compte de toutes les situations non coopératives. Ce processus de raisonnement, basé sur les croyances, est totalement distribué à l'intérieur des agents. Il n'y a pas d'analyse globale du travail.

Nous avons effectué des tests sur 30 configurations de "tileworld" : tous les mondes étant composés d'une combinaison de 2 à 5 agents, de 1 à 4 trous, de 1 à 6 pavés et de 2 à 10 barrières. Nous avons calculé le nombre d'étapes où chaque agent raisonne en poussant autant de pavés que possible dans les trous répartis sur une grille 10*10.

Nous avons étudié notre méthode sur 5 scénarios.

- Scénario **individualiste**. Le niveau de coopération de chaque agent est 0. Un agent est seulement intéressé par ses objectifs individuels. Ceci est possible parce qu'un agent peut faire seul tous les travaux pour terminer la session.

- Scénario **non ambigu**. Un agent n'est capable de traiter que les ambiguïtés.

- Scénario **non concurrent**. Un agent n'est capable de traiter que les situations de concurrence.

- Scénario **non conflictuel**. Un agent n'est capable de traiter que les conflits.

- Scénario **coopération totale**. Le niveau de coopération de chaque agent est optimal. Chaque agent est capable de considérer et de résoudre l'ambiguïté, la concurrence, et les situations de conflit. En fait, les agents essaient, s'ils le peuvent, d'aider ou d'être aidés par le collectif.

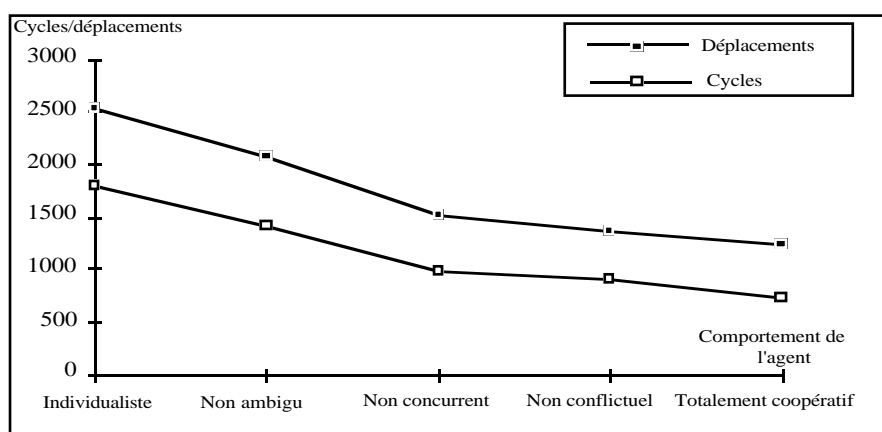


Figure A.2.2.a. : Nombre de cycles/déplacements selon le comportement de l'agent

Nous avons tout d'abord remarqué que la terminaison a toujours été obtenue avec des stratégies de coopération totale, ce qui n'est pas le cas avec des attitudes de coopération partielles et a fortiori lorsque les agents sont totalement individualistes.

Dans toutes les sessions, nous avons également noté que la coopération totale (résolvant successivement l'ambiguïté, la concurrence et le conflit) donnait de meilleurs résultats en nombre de déplacements et de cycles que les agents individuels (Cf. courbes de la figure A.4.2.a.).

Les résultats obtenus montrent aussi qu'une attitude coopérative basée sur l'aide mutuelle n'est intéressante que si les agents sont capables de se coordonner. Les résultats sont alors meilleurs que les résultats obtenus avec les agents individuels. La caractéristique générale qui émane de ces sessions est mise en évidence dans les courbes de la figure A.4.2. : plus les agents sont coopératifs, meilleure est l'organisation de la société et plus les agents sont capables de faire face à des situations imprévues donc de s'adapter.

La courbe des cycles indique clairement que la coopération peut s'adapter à un raisonnement totalement distribué chez les agents, car les agents ont une activité parallèle. Si ce n'était pas le cas, ils auraient fréquemment des interblocages dans leurs déplacements et le nombre de cycles ne serait pas aussi faible.

Méthode de coopération	Coopération bénéfique	Coopération neutre	Coopération nuisible
Goldman	17,56 %	77,02 %	5,40 %
Non cognitif	66,67 %	33,33 %	0 %

Figure A.2.2.b. : Comparaison avec les résultats de Goldman

Les résultats inscrits dans la tableau A.2.2.b. mettent en évidence le fait que notre stratégie est plus efficace à 66% qu'une stratégie avec des agents individuels. La coopération totale est donc moins utile dans 33% des cas.

Pour Goldman et Rosenschein [Goldman, 1994], la coopération est mise en oeuvre grâce à des règles de changement d'états (ou méta-règles) qui guident le comportement des agents. Dans le domaine du "tileworld", par exemple, une de ces méta-règles consiste à avoir des pavés pouvant bouger librement (ayant un degré de liberté maximum). L'auteur donne également différents niveaux de coopération entre agents. Les résultats de leurs sessions montrent, comme dans notre application, qu'un niveau de coopération positif est bénéfique pour le comportement de la société. Pour terminer avec une comparaison, les trois attitudes que nous avons définies et qui dictent le comportement que doit adopter un agent face à une situation non coopérative peuvent être vues comme des méta-règles du système de Goldman. Le niveau de coopération peut donc être réalisé en dotant les agents d'attitudes plus ou moins coopératives.

A.3. APPLICATION AUX FOURRAGEMENTS PAR DES FOURMIS

Benoît Morisset [Morisset, 1997], a cherché à instancier les quatre classes de situations non coopératives décrites au paragraphe 5.2. dans le cadre de sociétés de fourmis en phase de fourragement. A chaque détection d'un état non coopératif, il a associé une règle comportementale permettant de rétablir une situation coopérative. Il a pour cela effectué deux simulateurs : un premier (modèle naturel) permettant, à partir d'observations fournies par des éthologues, de reproduire le plus fidèlement possible le comportement de fourmis en phase de fourragement, puis un second dans lequel le comportement des fourmis est guidé par un comportement coopératif (modèle artificiel).

Chaque simulation est située dans un environnement paramétrable [Salles, 1997] comprenant un certain nombre d'obstacles, d'items de nourritures, de nids et de fourmis. L'environnement obéit à certaines lois physiques notamment pour assurer l'évaporation des cellules de phéromones. Les fourmis, quant à elles, sont capables de percevoir leur environnement de façon locale : aucune connaissance globale n'est permise. Les fourmis peuvent identifier une source de nourriture, une piste de phéromone, une autre fourmi, le nid ou un obstacle. Chaque élément de l'environnement perçu entraîne une réaction : la nourriture et la phéromone attirent la fourmi selon des forces paramétrables, un obstacle la repousse... Aucune communication directe par envoi de message n'est autorisée conformément à la réalité biologique. Le seul moyen de communication disponible pour les fourmis est le dépôt de phéromone : celui ci trace un chemin perceptible reliant le nid à une source de nourriture exploitée.

A.3.1. Détection et traitement des situations non coopératives

Seulement deux des situations non coopératives définies au paragraphe 5.2 ont pu être instanciées car l'incompétence, l'inutilité et le conflit ne sont pas pertinentes pour l'application. Pour chaque situation non coopérative, nous présentons comment un agent la détecte puis la façon de l'instancier dans le domaine d'application.

A.3.1.1. La situation non coopérative d'ambiguïté

Une fourmi est en situation d'ambiguïté lorsqu'elle perçoit à la fois une piste de phéromone et une source de nourriture non encore exploitée.

Dans un tel contexte, la fourmi opte pour l'exploitation de la source de nourriture puisque celle-ci n'est pas encore repérée. L'exploitation de cette source permet de la relier au nid par une piste

chimique ce qui permettra à d'autres fourmis de venir l'exploiter. Cette réaction a pour effet d'accroître le niveau d'information utile à l'ensemble de la colonie.

A.3.1.2. La situation non coopérative de concurrence

La situation de concurrence est mise en évidence dans deux cas de figure :

- lorsqu'une fourmi se déplace dans le but d'exploiter une source de nourriture déjà colonisée par beaucoup de fourrageuses. Si, dans la position où elle est, elle perçoit un autre item de nourriture (peut-être beaucoup plus petit mais néanmoins inexploité) elle a un choix concernant l'action qu'elle peut entreprendre. Cette situation est très fréquente lorsque l'environnement contient beaucoup d'items de nourriture mais de très faible granularité.

La fourmi va alors rétablir une situation coopérative en choisissant de préférence un item inexploité. Ainsi, en revenant au nid, elle va laisser un nouveau chemin de phéromone qui permettra à ses congénères de localiser la nouvelle ressource.

- lorsque deux fourmis en mode exploration se rencontrent. Cette rencontre signifie que les deux exploratrices fouillent le même espace. Il y a donc de fortes chances pour que la nourriture présente dans un périmètre proche de leur position soit perçue par les deux fourmis en même temps, ce qui a pour effet d'engendrer une situation concurrentielle.

Pour éviter cette situation non coopérative, chaque fourmi repousse l'autre. Ce comportement permet l'exploitation de zones dont l'intersection est quasiment vide. En s'éloignant ainsi l'une de l'autre, les deux fourmis vont exploiter le plus largement possible leur environnement et optimiser ainsi la localisation de ressources de nourriture.

A.3.1.3. La situation non coopérative d'improductivité

Lorsqu'une fourmi retourne au nid en laissant un message phéromonal et perçoit une source de nourriture non exploitée (elle ne perçoit pas de congénère), le comportement naturel voudrait que son comportement reste inchangé : sa charge est maximale, elle ne peut plus exploiter, donc elle rentre au nid et elle ne signale pas cette source. Or, en agissant de la sorte, elle n'est pas en situation coopérative avec ses congénères. En effet, elle a détecté une information qui ne lui est certes pas utile mais qui peut être profitable à autrui.

Dans un tel contexte, la fourmi va modifier l'intensité du message chimique à la hauteur de la source perçue. Ce renforcement encouragera ensuite le recrutement aux abords du nid. Lorsque les premières fourmis remontant la piste chimique renforcée, arrivent à la hauteur de la source, elles ne manqueront pas de la percevoir et se trouveront alors en situation d'ambiguïté ou de concurrence.

A.3.2. Résultats expérimentaux et analyse

Les résultats obtenus suite à cette simulation ont été comparés au modèle naturel simulé. Il est apparu que d'une manière générale, les performances réalisées par le système artificiel se sont révélées meilleures que celles établies par le modèle naturel. Cet écart de performance s'est accentué proportionnellement au niveau de difficulté imposé par l'environnement. Par exemple, plus les sources sont éloignées, plus le fourrage nécessite une bonne organisation et plus l'écart entre les deux systèmes s'accroît à l'avantage du modèle artificiel. Ceci montre que le principe d'auto-organisation développé engendre des systèmes capables de s'adapter à une large plage de conditions (en jouant sur les paramètres de l'environnement) et que les situations difficiles, nécessitant des stratégies performantes, révèlent une véritable efficacité. Le détail des résultats peut être consulté dans [Morisset, 1997].

Nous allons étudier un résultat obtenu avec deux sociétés (une simulant une société naturelle, l'autre étant coopérative) comprenant respectivement 25 fourmis, chaque fourmi pouvant transporter jusqu'à 10 unités de nourriture et ayant un champ de perception de $20 * 20$ pixels. L'environnement comprend 65 items de nourriture de taille 3 à 40. La fréquence d'évaporation de la phéromone est de 50 cycles, et enfin, la durée maximum pendant laquelle la fourmi est autorisée à rester à l'extérieur du nid est de 800 cycles. Voici les résultats obtenus avec une simulation effectuée durant 2000 cycles :

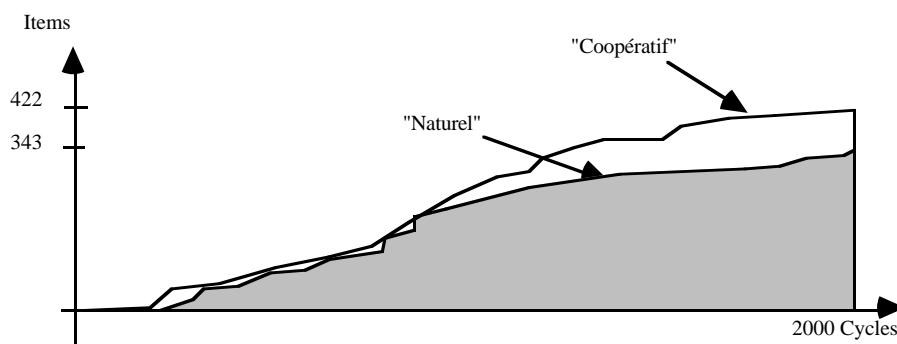


Figure A.3.2. Résultat expérimental

On voit que le nombre d'items de nourriture découvert par le système coopératif est globalement supérieur (18,7%) à celui obtenu par le système naturel.

Le début de la récolte démarre plus tôt pour le système artificiel. La pente des courbes est dans un premier temps comparable. Nous pouvons ensuite observer une seconde phase dans laquelle la différence entre les deux systèmes est plus importante. Les deux stratégies de récolte engendrent une accélération, mais celle-ci est nettement plus prononcée pour le système artificiel. A la sortie de la première phase les deux systèmes ont accompli sensiblement les mêmes performances quantitatives à savoir l'exploitation des sources proches du nid (facilement repérables). La

différence se juge en termes qualitatifs car le système artificiel est mieux organisé à la sortie de la première phase. Les ressources repérées sont en plus grand nombre grâce à la suppression des situations concurrentielles. L'exploitation massive de ces ressources lors de la phase suivante explique le décalage existant.

A.4. CONCLUSION

Pour faire face à des situations non coopératives telles que les conflits, les ambiguïtés, les concurrences ou les incompétences, nous avons doté chaque agent d'attitudes sociales coopératives couplées à des mécanismes généraux de détection et de traitement de ces situations qui lui permettent de s'adapter. Le traitement effectué par l'agent dans ces cas là fait en sorte qu'il se retrouve dans une situation coopérative.

A l'aide des deux exemples présentés, nous avons pu montrer que des agents coopératifs pouvaient simultanément satisfaire la tâche individuelle qui leur était assignée tout en ayant une activité collective meilleure que celle d'agents individualistes. En effet, les agents n'étaient pas "programmés" dans le but d'atteindre un but collectif particulier : la fonction observable peut donc être qualifiée d'émergente car elle est réductible à la dynamique individuelle. Ces deux applications illustrent donc bien tout le profit que l'on peut tirer d'attitudes coopératives pour augmenter la performance globale d'un système.

Cependant, les agents modélisés à la fois dans le monde du "tileworld" et dans celui des "fourmis" ne sont pas auto-organiseurs. En effet, ils ne modifient ni leurs compétences, ni leurs croyances, ni leurs attitudes sociales. Ils s'adaptent aux perturbations et cette adaptation n'est que momentanée, elle n'est pas mémorisée.

Pour qu'ils puissent s'auto-organiser, les agents doivent être capables d'apprendre. Cet apprentissage doit être décidé de manière autonome par chaque agents. Le principe est le suivant : chaque agent doit se positionner dans l'organisation de manière à être en situation permanente d'interaction coopérative avec les autres membres du collectif : l'organisation doit être optimale. Cet apprentissage par auto-organisation (reposant toujours sur la recherche permanente de coopération avec autrui) est en revanche présent dans le projet ARCADIA.

Cette annexe présente la trace obtenue dans le cadre de l'exemple présenté au paragraphe 11.2.3. qui met en évidence la communication entre AdT et AdM ainsi que la communication spontanée entre AdTs.

Les différents AdTs présents sur AdM0 (AdT1 et AdT4) sont respectivement appelés marc v (vendeur) et paula (acheteur) dans les traces fournies.

Le prototype possède une fenêtre par processus afin de permettre une meilleure visibilité des communications inter-processus. Nous avons conservé cette présentation : un listing est fourni par processus.

Seuls les numéros ont été rajoutés pour permettre au lecteur de suivre l'évolution des communications schématisées dans la figure 11.2.3. que nous rappelons :

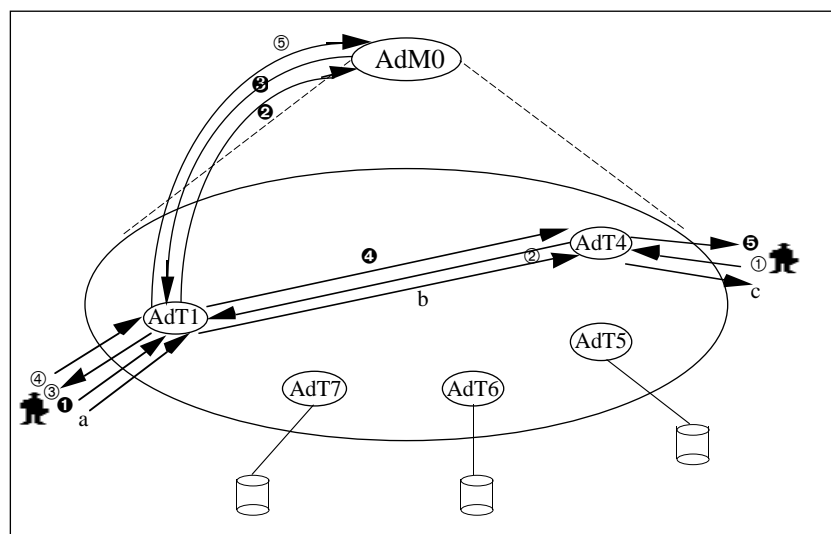


Figure B.1. : La communication spontanée (réalisée lors de la communication b)

Cette annexe présente la trace obtenue dans le cadre de l'exemple présenté au paragraphe 11.4.2. qui met en évidence à la fois la communication inter-AdMs, la relaxation entre AdMs et la relaxation entre AdTs.

Voici les correspondances entre les AdMs et AdTs cités dans le paragraphe 11.4.2. et ceux présents sur la trace :

AdM0 \longleftrightarrow AdM0
 AdT1 \longleftrightarrow marc
 AdM2 \longleftrightarrow AdM2
 AdM1 \longleftrightarrow AdM1
 AdT1 \longleftrightarrow AdT1
 AdT3 \longleftrightarrow pavehicules
 AdT4 \longleftrightarrow paula

Le prototype possède une fenêtre par processus afin de permettre une meilleure visibilité des communications inter-processus. Nous avons conservé cette présentation : un listing est fourni par processus excepté pour AdM1 et AdT1 (AdT1 est un agent émissaire qui est créé par AdM1) qui partagent le même listing.

Seuls les numéros ont été rajoutés pour permettre au lecteur de suivre l'évolution des communications schématisées dans la figure 11.2.3. que nous rappelons :

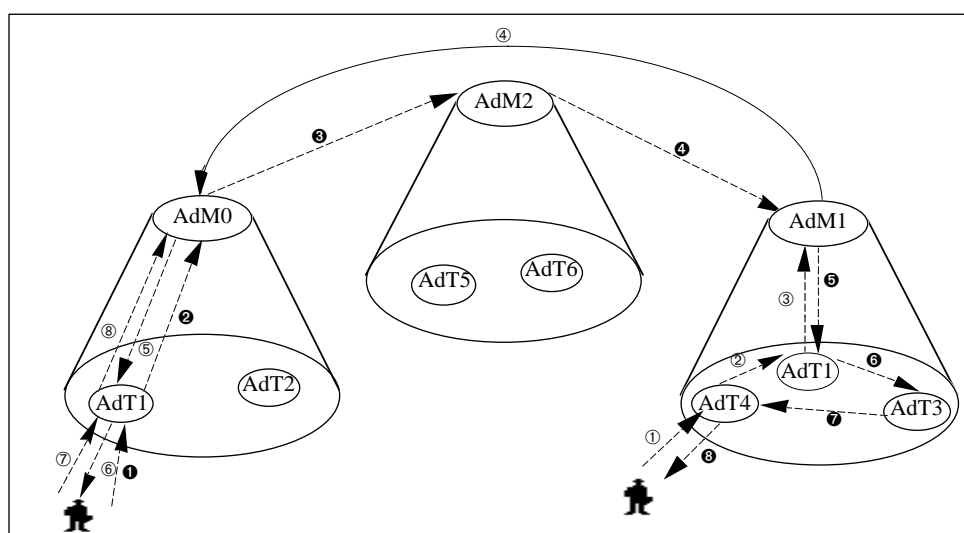


Figure C.1. : Mise en évidence de la relaxation restreinte au niveau des AdMs (cas ④)