INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents

S. Labidi, W. Lejouad

N° 2004

Août 1993

PROGRAMME 2

Calcul symbolique,
programmation
et génie logiciel

Rapport de recherche

1993

ISSN 0249-6399

De l'Intelligence Artificielle Distribuée aux Systèmes Multi-Agents

S. Labidi, W. Lejouad

Programme 2 — Calcul symbolique, programmation et génie logiciel
Projet SECOIA

Rapport de recherche n° 2004 — Août 1993 — 39 pages

Résumé : Les recherches en intelligence artificielle distribuée (IAD) et systèmes multi-agents (SMA) étudient la manière de répartir un problème sur un certain nombre d'entités coopérantes. Elles étudient aussi la manière de coordonner le comportement intelligent d'un ensemble d'entités selon des lois sociales. Ces entités ou *agents* sont autonomes et interagissent dans un environnement pour la résolution de problème. L'aspect collectif de ces agents nécessite l'étude de nouveaux concepts et de nouvelles théories qui émergent d'une *société d'agents*. Ces thèmes de recherche sont en cours d'exploration, l'objectif de ce rapport est de donner une vue d'ensemble sur leur évolution.

Mots-clé : IA Distribuée, Systèmes Multi-Agents et Systèmes Ouverts.

(Abstract: pto)

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

From Distributed Artificial Intelligene to Multi-Agent Systems

Abstract: Distributed Artificial Intelligence (DAI) and Multi-Agent Systems (MAS) consider how problem solving can be divided among several cooperating entities. They are also concerned with coordinating intelligent behavior among a collection of entities in respect to some social laws. These entities or *agents* are autonomous and interact in an environment to solve problems. The collective aspect of these agents requires the definition of new concepts and theories that emerge in an *agent society.* Such research topics are in progress, the purpose of this report is to give an overview on their evolution.

Key-words: Distributed AI, Multi-Agent Systems and Open Systems.

1 Introduction

À la différence de l'Intelligence Artificielle classique (IA) qui modélise le comportement intelligent d'un seul agent, l'intelligence artificielle distribuée (IAD) s'intéresse à des comportements intelligents qui sont le produit de l'activité *coopérative* de plusieurs agents.

Le passage du comportement individuel aux comportements collectifs est considéré non seulement comme une extension mais aussi comme un enrichissement de l'IA, d'où émergent de nouvelles propriétés et de nouveaux comportements.

L'objectif de ce rapport est de donner une vue d'ensemble sur les recherches en IAD et les disciplines qui en découlent, en l'occurrence, les Systèmes Multi-Agents (SMA).

Dans ce rapport, nous commençons par introduire l'IAD et présenter quelques modèles de base de ses systèmes, ensuite nous mettons l'accent sur l'évolution de cette discipline vers les SMA et nous décrivons les propriétés et lois qui émergent en passant de l'aspect individuel (le comportement d'un agent) à l'aspect collectif (son comportement dans une société d'agents), enfin nous présentons quelques applications de systèmes d'IAD pour illustrer l'état actuel de la pratique dans le domaine.

1.1 L'intelligence artificielle distribuée

L'évolution des domaines d'application de l'IA pour recouvrir des domaines complexes et hétérogènes tels que l'aide à la décision, la reconnaissance et la compréhension des formes, la conduite des processus industriels, etc., a montré les limites de l'approche classique de l'IA qui s'appuie sur une centralisation de l'expertise au sein d'un système unique.

Les travaux menés au début des années 70 sur la concurrence[1] et la distribution ont contribué à la naissance d'une nouvelle discipline : l'Intelligence Artificielle Distribuée (IAD) [11] [49] [38]. L'IAD a pour but de remédier aux insuffisances de l'approche classique de l'IA en proposant la distribution de l'expertise sur un groupe d'agents devant être capables de travailler et d'agir dans un environnement commun et résoudre les conflits éventuels. D'où la naissance de notions nouvelles en IA, telles que la coopération, la coordination d'actions, la négociation et l'émergence.

L'IAD peut alors être définie comme étant la branche de l'IA qui s'intéresse à la modélisation de comportement "intelligent" (jusque là c'est la définition de l'IA classique) par la coopération entre un ensemble d'agents [49].

1. La concurrence fait référence au déroulement logique simultané d'actions par opposition au parallélisme qui est le déroulement physique simultané d'actions.

1.2 Historique

Les besoins en parallélisme et distribution ont été très tôt ressentis en IA. En 1973, le premier système d'IAD a pu voir le jour. Il s'agit du système HEARSAY [31] pour la reconnaissance de la parole basé sur l'architecture de *blackboard* (cf. § 2).

Dans les recherches qui ont suivi, diverses approches liées à la distribution se sont succédées, notamment : *les acteurs* de Hewitt [45] (cf. § 3), *la société de l'esprit* de Minsky [61], le système DVMT de Lesser [54], le *Contract Net* de Smith [73], le système *Mace* de Gasser [41] et les *micro-robots* de Brooks [16]. Nous reviendrons en détail sur quelques-unes de ces approches par la suite (cf. § 2).

1.3 Thèmes de recherche de l'IAD

Après avoir défini l'IAD et la différence entre celle-ci et l'IA classique, nous présentons trois axes fondamentaux dans la recherche en IAD :

- *Les systèmes multi-agents* (SMA) [29] : Il s'agit de faire coopérer un ensemble d'agents dotés d'un comportement intelligent et de coordonner leurs buts et leurs plans d'actions pour la résolution d'un problème. C'est le thème auquel nous nous intéressons le plus dans ce rapport.

- *La résolution distribuée des problèmes* (RDP) [55] : Elle s'intéresse à la manière de diviser un problème particulier sur un ensembles d'entités distribuées et coopérantes. Elle s'intéresse aussi à la manière de partager la connaissance du problème et d'en obtenir la solution.

- *L'Intelligence artificielle parallèle* (IAP) [22] [32] : Elle concerne le développement de langages et d'algorithmes parallèles pour l'IAD. L'IAP vise l'amélioration des performances des systèmes d'intelligence artificielle sans, toutefois, s'intéresser à la nature du raisonnement ou au comportement intelligent d'un groupe d'agents. Cependant, il est vrai que le développement de langages concurrents et d'architectures parallèles peut avoir un impact important sur les systèmes d'IAD.

1.4 Problématique de l'IAD

Les problèmes que l'IAD s'attache à résoudre peuvent être divisés en deux classes : la première concerne les problèmes classiques de l'IA qui ont pris une nouvelle dimension dans le contexte multi-agents. La seconde regroupe les nouveaux problèmes proprement liés au thème de l'IAD. Dans la première classe on peut citer :

- La modélisation de la connaissance et le problème de sa répartition sur les différents agents : comment formuler, décomposer, allouer des problèmes et synthétiser les résultats d'un groupe d'agents ?
- Les problèmes de génération de plans d'actions où il faut prendre en considération la présence d'autres agents ;
- La gestion des conflits entre les agents (points de vue différents des agents) et le maintien de la cohérence des décisions et des plans d'actions ;
- Le problème de la communication : comment permettre la communication et l'interaction entre les agents ? Quel langage et quel protocole faut-il employer ? Une communication dans les univers multi-agents n'est plus une simple tâche d'entrée-sortie, mais doit être modélisée comme un acte pouvant influer sur l'état des autres agents.

Les problèmes de la seconde classe peuvent être divisés en deux :

- Les problèmes spécifiques au groupe d'agents, qui portent sur l'organisation, l'architecture de l'ensemble des agents et les paradigmes de coopération et d'action ;
- Les problèmes liés au comportement d'un agent au sein d'un groupe. On s'intéresse aux capacités sociales d'un agent : le partage des tâches, le partage des ressources, le raisonnement sur les autres agents (pouvoir modéliser leurs connaissances et être en mesure de connaître leurs plans d'actions et de raisonner en fonction de ces plans).

D'autres thèmes de recherche sont présents dans le contexte multi-agents, à savoir, le raisonnement temporel, le raisonnement hypothétique, la représentation de la connaissance imprécise, etc.

2 Modèles de systèmes d'IAD

Dans cette section, nous présentons quelques modèles qui ont marqué l'histoire de l'IAD.

Blackboard

Le modèle de *blackboard* [44] définit une architecture qui organise la résolution de problèmes par coopération de plusieurs modules, appelés *sources de connaissances* (KS), autour d'une base de données partagée appelée *blackboard*[2]. Chaque KS vient lire ou écrire sur le blackboard. D'une façon générale, le rôle d'une KS est de résoudre un sous-problème particulier en fonction de l'état du blackboard. Ce modèle vise la mise en œuvre d'une résolution opportuniste, où le choix de la KS à activer est déterminé en fonction des critères de contrôle actifs.

Contract Net

Le *Contract Net* [9] est un système de résolution de problème distribué, conçu par R. Davis et R. Smith. L'objectif principal de ce système étant la distribution, il procède par allocation des tâches à un ensemble de résolveurs de problèmes et utilise le concept de *négociation* pour adjuger des *contrats*. L'architecture de base contient des nœuds ayant des rôles de *chef* et d'*exécutants*.

DVMT

Distributed Vehicle Monitoring Testbed est un système d'analyse de trafic routier par synthèse des observations d'un certain nombre de capteurs. Ce système fournit un exemple d'application dans lequel la perception et la connaissance sont distribuées. DVMT utilise une collection de systèmes identiques à base de *blackboard* pour résoudre des problèmes d'interprétation des données à partir d'un ensemble de capteurs séparés géographiquement et couvrant une région. Les données sont acheminées d'un sous-ensemble de capteurs à destination de chaque nœud participant à la résolution du problème. Les nœuds coopèrent pour construire une image globale du trafic routier.

De nombreuses architectures sont insatisfaisantes quant à leur aspect centralisé, en particulier les modèles de blackboard qui donnent une vision globale du problème sous forme d'une base de données partagée. Ce sont les travaux sur le modèle *acteur* détaillé dans la section suivante, qui ont abouti à une nouvelle conception des systèmes d'IAD.

2. Le terme blackboard (ou tableau noir) désigne à la fois le modèle et la base partagée.

3 Modèle acteur

Le modèle acteur [46] est développé au MIT par l'équipe de C. Hewitt[3]. À l'origine, ce modèle a été conçu pour résoudre des problèmes d'IA, il a abouti à une conception des systèmes distribués, en l'occurrence, les systèmes ouverts. Le modèle acteur est un modèle sous-jacent aux systèmes ouverts.

3.1 Systèmes ouverts

Ce sont des systèmes devant fonctionner de la manière la plus indépendante possible. Selon Hewitt [47], les systèmes ouverts (SO) sont des systèmes distribués, ayant les caractéristiques suivantes :

- ils sont concurrents et asynchrones, avec un contrôle décentralisé ;
- ils sont composés d'entités indépendantes en évolution continue ;
- ils sont facilement extensibles ;
- chaque entité ne possède qu'une vision partielle du système ; elle doit négocier avec les autres pour coopérer.

3.2 Acteurs

Le modèle acteur est une extension du modèle objet par ajout de la notion d'activité. Dans les modèles objets, le modèle de calcul est un transfert de contrôle de l'objet expéditeur à l'objet receveur de message, ce qui signifie que l'activité principale est l'allocation du processeur à un objet. Un seul objet est donc actif à la fois et ce sont les messages qui activent les objets. Au contraire, dans le modèle acteur le modèle de calcul est un transfert d'information (requête) entre processus concurrents.

Dans ce modèle, un acteur est un objet actif, autonome, communiquant par envoi de messages asynchrones avec les autres acteurs. Les langages d'acteurs permettent ainsi de garder les acquis des langages à objets tout en y ajoutant la concurrence (coopération entre plusieurs objets actifs simultanément) ; on parle alors de programmation orientée objet concurrente (POOC) [1]. Un acteur se compose d'un ensemble limité de connaissances : les acteurs qu'il connaît directement appelés ses *accointances* et un comportement défini par un *script*. Un *script* est un ensemble de méthodes décrivant la manière dont il doit répondre aux événements venant de l'extérieur : l'acteur filtre les messages reçus et active les méthodes appropriées.

3. Devenue depuis le Message Passing Semantics Group.

On distingue deux types d'acteurs : les acteurs *sérialisés* et les acteurs *non sérialisés*. Les acteurs *sérialisés* sont les seuls à pouvoir changer d'état suite à l'exécution d'un message, ils ne traitent qu'un message à la fois et spécifient explicitement leur nouvel état avant de traiter le message suivant. Les acteurs *non sérialisés* peuvent traiter un nombre arbitraire de messages en parallèle [58].

3.3 Communications

L'activité d'un acteur consiste à recevoir des messages et à élaborer des réponses. Un acteur peut *déléguer* les messages auxquels il ne peut pas répondre. Dans les sections suivantes, nous présentons les communications dans un modèle acteur.

3.3.1 Envoi de messages

Dans un monde d'acteurs, tout est acteur et le seul événement qui peut se produire est l'envoi de message : un acteur émetteur transmet à un acteur récepteur un acteur message. Ceci se fait normalement d'une manière asynchrone, mais certains langages d'acteurs (comme ABCL/1 [81]) offrent aussi les communications synchrones ou anticipées[4].

Les messages contiennent des *continuations* qui sont des informations supplémentaires permettant de désigner l'acteur auquel sera envoyé le résultat du message. Une continuation peut être un autre acteur ou un acteur créé spécialement pour recevoir ce résultat. Par défaut, la continuation est l'émetteur du message.

3.3.2 Délégation

Il s'agit d'un mécanisme permettant à chaque acteur ne pouvant pas traiter un message de le *déléguer* à un autre acteur appelé *proxy*: le *proxy* prend en charge l'exécution du message et l'acteur déléguant est immédiatement disponible pour exécuter d'autres messages. Le *proxy* connaît l'acteur déléguant pour le cas où il aurait besoin d'informations supplémentaires.
Pour éviter une régression à l'infini dans le processus de délégation, il y a des acteurs qui ne délèguent jamais : ce sont les acteurs primitifs (appelés *rock-bottom* dans des langages comme Act1 [57]).

4. Une communication anticipée permet à un acteur d'envoyer un message dont il n'utilisera le résultat qu'ultérieurement : l'acteur continue son activité et sollicitera le résultat au moment où il en aura besoin, et si le résultat n'est pas encore disponible, il se mettra en attente.

La délégation est un moyen de partage de connaissances et de comportements plus souple que la notion d'héritage : un acteur choisit dynamiquement son *proxy* alors que le partage de connaissance décrit par un graphe d'héritage est statique. La délégation se caractérise donc par son aspect asynchrone, dynamique et non bloquant.

3.4 Le modèle de Gul Agha

Gul Agha a donné une définition formelle du modèle acteur tout en étendant sa sémantique : un acteur est désormais défini par l'adresse de sa boîte aux lettres où seront envoyés les messages qui lui sont destinés, et un comportement définissant le traitement de ces messages. Il a aussi ajouté le mécanisme de délégation qui n'existait pas chez Hewitt. Au niveau de la communication entre acteurs, il n'a retenu que la transmission asynchrone, toutefois les transmissions synchrones et anticipées peuvent s'exprimer en terme de transmissions asynchrones [82].

3.5 Conclusion

Les systèmes d'acteurs sont caractérisés par la communication par envoi de messages et un traitement local : un agent (acteur) détient des informations locales, et son comportement est indépendant de celui des autres. Ils présentent une distribution à la fois des connaissances, des résultats partiels et des méthodes utilisées pour la résolution du problème. En revanche, le comportement global d'un système d'acteurs est difficile à analyser, et les langages d'acteurs restent peu diffusés et sont loin d'être des produits finis. Les travaux menés dans ce sens tiennent plus de la conception de langages que de systèmes de résolution de problèmes [5].

Une nouvelle approche en IAD se développe actuellement, elle essaye de faire coopérer des entités auxquelles sont rattachées des caractéristiques de haut niveau, tout en essayant de respecter les principes des systèmes ouverts. Ces entités seront désormais nommées *agents* et les systèmes correspondants seront appelés *systèmes multi-agents*. Nous étudions, dans ce qui suit, l'univers des systèmes multi-agents.

4 Évolution vers les systèmes multi-agents

Contrairement à l'approche centralisée de l'IA, l'intelligence artificielle distribuée vise la distribution de l'expertise sur un ensemble de composants qui

5. Peut-être, avec l'arrivée des architectures parallèles et des réseaux de processeurs, prendront-ils une autre dimension ?

communiquent pour atteindre un objectif global (élaboration d'un diagnostic, résolution d'un problème, etc.). Cette approche nécessite la division du problème en sous-problèmes. Mais cette hypothèse n'est pas toujours réaliste car beaucoup de problèmes ne peuvent être partitionnés de cette manière [42].

Une extension des systèmes d'IAD est proposée : les composants doivent être capables de raisonner sur les connaissances et les capacités des autres dans le but d'une coopération effective [50]. Pour ce faire, ils doivent être dotés de capacités de perception et d'action sur leur environnement et doivent posséder une certaine autonomie de comportement, on parle alors d'agents[6] et par conséquent de systèmes multi-agents [26].

Les systèmes multi-agents se caractérisent alors par l'autonomie et l'intelligence des composants impliqués. Toutefois, un agent ne dispose pas d'une vision globale de son environnement.

4.1 Concept d'agent

Un agent peut être défini comme une entité (physique ou abstraite) capable d'agir sur elle-même et son environnement, disposant d'une représentation partielle de cet environnement, pouvant communiquer avec d'autres agents et dont le comportement est la conséquence de ses observations, de sa connaissance et des interactions avec les autres agents [38].

Les agents ont deux tendances : une *tendance sociale* tournée vers la collectivité (les mécanismes et connaissances associés concernent les activités du groupe) et une *tendance individuelle* avec des mécanismes et des connaissances contenant les règles de fonctionnement interne de l'agent. On peut caractériser un agent par : son rôle, sa spécialité, ses objectifs et ses fonctionnalités, ses croyances, ses capacités décisionnelles, ses capacités de communication et éventuellement ses capacités d'apprentissage. Nous détaillons l'architecture et le fonctionnement d'un agent dans le § 5.

4.2 Déterminant d'un agent

On appelle *déterminant* d'un agent l'ensemble nécessaire et suffisant de ses caractéristiques structurelles, environnementales et comportementales qui permet d'expliquer ses façons d'agir.

Les caractéristiques environnementales d'un agent sont liées à la représentation que se fait l'agent de son environnement et de lui-même. Les caractéristiques structurelles d'un agent déterminent l'ensemble de ses composants, alors que les caractéristiques comportementales contraignent l'ensemble

6. La terminologie dans le domaine n'est pas encore clairement définie par la communauté IAD, et le terme agent reste encore un terme générique.

de ses comportements, en accord avec les caractéristiques environnementales. Par exemple, un avion est un agent dont les caractéristiques structurelles sont la capacité, la vitesse, etc. Les caractéristiques comportementales sont le pilote et le plan de vol et sa caractéristique environnementale est sa position [59].

4.3 Agents cognitifs vs réactifs

La granularité[7] des agents impliqués dans une application varie selon deux écoles coexistantes aujourd'hui : l'école cognitive et l'école réactive. Suivant le type d'agent utilisé, on parlera de systèmes cognitifs ou de systèmes réactifs.

4.3.1 Agents cognitifs

Les systèmes d'agents cognitifs sont fondés sur la coopération d'agents capables à eux seuls d'effectuer des opérations complexes. Un système cognitif comprend un petit nombre d'agents qui disposent d'une capacité de raisonnement sur une base de connaissances, d'une aptitude à traiter des informations diverses liées au domaine d'application, et d'informations relatives à la gestion des interactions avec les autres agents et l'environnement [24]. Chaque agent est assimilable, suivant le niveau de ses capacités, à un système expert plus ou moins sophistiqué, on parle d'agent de forte granularité (*coarse grain*).

4.3.2 Agents réactifs

Les défenseurs de cette approche partent du principe suivant : *dans un système multi-agents, il n'est pas nécessaire que chaque agent soit individuellement "intelligent" pour parvenir à un comportement global intelligent.* En effet, des mécanismes simples de réactions aux événements peuvent faire émerger des comportements correspondant aux objectifs poursuivis. Cette approche propose la coopération d'agents de faible granularité (*fine grain*) mais beaucoup plus nombreux.

Les agents réactifs sont de plus bas niveau, ils ne disposent que d'un protocole et d'un langage de communication réduits, leurs capacités répondent uniquement à la loi stimulus/action.

Les premiers travaux relatifs à cette approche ont été réalisés au MIT en 1986 par R. Brooks [16]. D'après lui, plusieurs milliers de micro-robots identiques, d'une taille aussi petite que possible, travaillant ensemble sur une tâche donnée pourront être plus efficaces qu'un seul gros robot spécialisé[8]. Dans le

7. On entend par *granularité* le degré de détail des connaissances de l'agent. La granularité exprime la complexité des fonctionnalités d'un agent.

8. La construction de ces petits agents simples et dotés d'une autonomie réelle ne sera pas plus difficile que la fabrication des circuits intégrés d'aujourd'hui !

Systèmes d'agents cognitifs	*Systèmes d'agents réactifs*
Représentation explicite de l'environnement	Pas de représentation explicite
Peut tenir compte de son passé	Pas de mémoire de son historique
Agents complexes	Fonctionnement stimulus/réponse
Petit nombre d'agents	Grand nombre d'agents

TAB. 1 - les agents cognitifs vs réactifs

même sens, une équipe de l'université de Bruxelles développe des sociétés de micro-robots destinés à l'utilisation en milieu hostile ou au recueil d'échantillon sur d'autres planètes [30]. Une fois lancés dans la nature, ces micro-robots doivent se débrouiller seuls, s'adapter à l'environnement rencontré et coordonner leurs actions. Si l'un d'entre eux tombe en panne, le relais est pris par d'autres et le fonctionnement global ne sera pas condamné à l'arrêt (ceci n'est pas le cas si on utilise un seul gros robot et qu'un de ses membres tombe en panne). Le nombre important de robots utilisés permet de supporter des pertes sans remettre en cause l'objectif final.

En France on peut citer les travaux de J. Ferber à l'Université de Paris VI, qui a travaillé sur le langage MERING[9] [33] [34]. Plusieurs versions se sont succédées depuis 1982, la dernière, MERING IV [37], est un langage concurrent pour la représentation des connaissances qui peut être utilisé comme une plate-forme pour la conception de systèmes multi-agents réactifs. D'autres travaux plus récents portent sur l'éco-résolution considérée comme exemple d'agents réactifs [36].

Conclusion

Dans le tableau 1, nous résumons les différences entre les deux approches. C'est l'école cognitive qui, jusqu'à maintenant, a donné lieu aux applications les plus avancées. La complexité des systèmes réactifs exige le développement de nouvelles théories dans le domaine de la coopération, de la communication et de la compréhension de nouveaux phénomènes telle que l'émergence. Toutefois, il est possible de concevoir des systèmes hétérogènes comportant les deux types d'agents. Il est aussi possible de doter les agents cognitifs de capacités de réactions aux événements : on parlera alors d'agents hybrides[10].

9. Premier langage hybride français, il a introduit la notion de réflexion.
10. À citer dans ce sens les travaux de l'équipe d'IA de l'université de Laval [18].

4.4 Caractéristiques d'un agent

L'avancement des travaux en IAD et SMA a conduit les chercheurs à définir non seulement la notion d'agent mais aussi quelques-unes de ses caractéristiques [11].

Intentionnalité

Un agent intentionnel est un agent guidé par ses buts. Une intention [70] [77] [69] est la déclaration explicite des buts et des moyens d'y parvenir. Elle exprime donc la volonté d'un agent d'atteindre un but ou d'effectuer une action. Notons que la notion d'intentionnalité n'existe pas chez Minsky [12].

Rationalité

Un agent rationnel est un agent qui suit le principe suivant (dit principe de rationalité) [63] :

> *"Si un agent sait qu'une de ses actions lui permet d'atteindre un de ses buts, il la sélectionne".*

Les agents rationnels disposent de critères d'évaluation de leurs actions, et sélectionnent selon ces critères les meilleures actions qui leur permettent d'atteindre le but. De tels agents sont capables de justifier leurs décisions.

La notion de rationalité se rapporte au comportement cognitif de l'agent. Ce terme qualifie l'utilisation efficace des ressources par l'agent. Les critères d'évaluation de cette rationalité sont étudiés dans [60].

Engagement

La notion d'engagement [6] [10] est l'une des qualités essentielles des agents coopératifs. Un agent coopératif planifie ses actions par coordination et négociation avec les autres agents. En construisant un plan pour atteindre un but, l'agent se donne les moyens d'y parvenir et donc s'engage à accomplir les actions qui satisfont ce but : l'agent croit qu'il est en mesure d'exécuter tout le plan qu'il a élaboré, ce qui le conduit (ainsi que les autres agents) à agir en conséquence.

Adaptativité

Un agent adaptatif est un agent capable de contrôler ses aptitudes (communicationnelles, comportementales, etc.) selon l'agent avec qui il interagit. Un agent adaptatif est un agent d'un haut niveau de flexibilité.

11. Les caractéristiques énoncées dans cette section se rapportent aux agents cognitifs.
12. Minsky utilise la notion d'*objectif* [61].

Intelligence

En conclusion, on appelle agent intelligent un agent cognitif (cf. § 4.3.1), rationnel, intentionnel et adaptatif.

5 Architecture d'agent

Dans ce paragraphe, nous présentons une architecture générale d'agent cognitif et nous décrivons les processus mis en œuvre lors de son fonctionnement (cf. figure 1 et figure 2).

5.1 Structure interne

La figure 1 décrit l'architecture globale d'un agent cognitif. C'est une synthèse des architectures d'agents décrites dans la littérature. On distingue essentiellement : le savoir-faire, les croyances, la connaissance de contrôle, l'expertise de l'agent et la connaissance de communication.

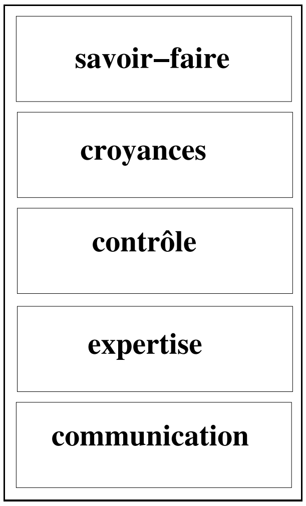

FIG. 1 - structure d'un agent

- *Savoir-faire* : Le savoir-faire est une interface permettant la déclaration des connaissances et des compétences de l'agent. Il permet la sélection des agents à solliciter pour une tâche donnée. Il n'est pas nécessaire mais il est très utile pour améliorer les performances du système, quel que soit le mode de coopération utilisé.

- *Croyances* : Dans un univers multi-agents, chaque agent possède des connaissances sur lui-même et sur les autres. Ces connaissances ne sont pas nécessairement objectives, on parle alors de croyances d'un agent. Les logiques des connaissances et des croyances [43] s'intéressent à la formalisation de telles connaissances considérées comme incertaines. Cette formalisation est à la base de la conception de tout système multi-agents puisqu'elle détermine en grande partie le comportement "intelligent" des agents.

- *Contrôle* : La connaissance de contrôle dans un agent est représentée par les buts, les intentions, les plans et les tâches qu'il possède (cf. § 4.4).
- *Expertise* : C'est la connaissance sur la résolution de problème. Pour un système expert utilisant le formalisme de règle, par exemple, cette connaissance correspond à sa base de règles.
- *Communication* : L'agent doit posséder un protocole de communication lui permettant d'interagir avec les autres agents pour une bonne coopération et une bonne coordination d'actions (cf. § 6.5). D'autres connaissances de communication peuvent être disponibles, par exemple les connaissances sur les réseaux de communication (tous les agents ne sont pas forcément en liaison directe).

À ces différents types de connaissances, on peut ajouter la connaissance liée au mode de la coopération : fonctionnement en mode *appel d'offre*, en mode *compétition* ou en mode *commande* (cf. § 6.3.2).

5.2 Fonctionnement

Dans ce paragraphe, nous présentons notre vision de l'architecture fonctionnelle d'un agent cognitif et nous détaillons son fonctionnement. Cette architecture est illustrée par la figure 2. Dans cette figure, les ellipses représentent les processus mis en œuvre lors du fonctionnement de l'agent.

Les agents sont immergés dans un environnement dans lequel et avec lequel ils interagissent. D'où leur structure autour de trois fonctions principales : *percevoir*, *décider* et *agir*. Parmi les sous-fonctions importantes d'un agent on peut citer : la détection de conflits, la révision des croyances, la coopération (négociation, coordination), l'apprentissage, etc. Toutes les fonctionnalités ne sont pas représentées dans la figure.

Un agent a la possibilité d'acquérir des connaissances sur l'environnement externe (*perception*). Il a aussi des capacités d'interaction avec les autres agents (communication, négociation). En fonction des connaissances et croyances dont il dispose et des buts qu'il se fixe suite à une perception ou à une interaction avec le monde extérieur, l'agent doit élaborer un plan d'action. Pour cela, il doit

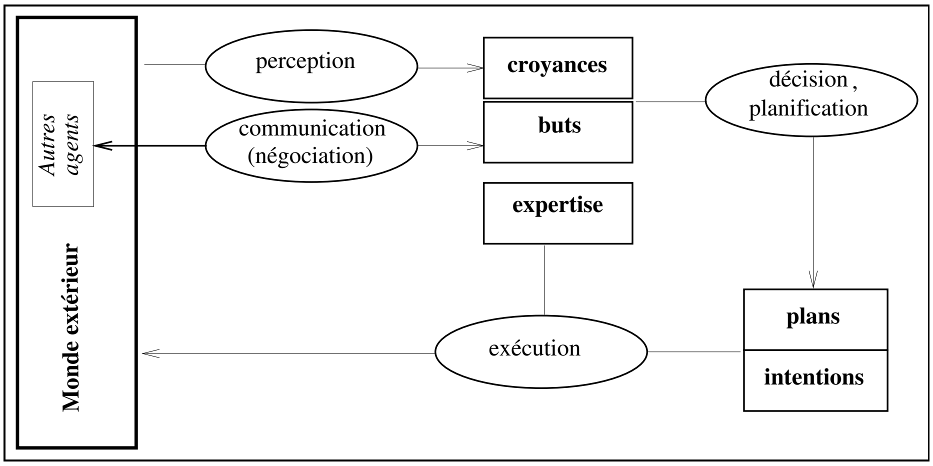

FIG. 2 - fonctionnement d'un agent

décider quel serait le but à retenir et à satisfaire en premier, ensuite planifier en fonction de ce but et passer à l'exécution. Ces deux derniers processus doivent être alternés du fait du caractère dynamique des environnements multi-agents (cf. § 5.2.3).

5.2.1 Perception

Les connaissances d'un agent ont plusieurs origines :

- le savoir initial de l'agent ;
- la perception de soi (perception *proprioceptive*) et du monde (perception *extéroceptive*) ;
- la communication avec les autres agents.

Généralement, les informations issues de la perception et du savoir initial de l'agent sont considérées comme des connaissances *certaines* puisqu'elles n'ont subi aucune mise à jour, alors que les connaissances provenant des autres agents sont considérées *incertaines* puisqu'elles évoluent sans que l'agent en soit forcément informé. On peut pour cela associer à chaque connaissance son origine afin d'en évaluer la crédibilité et d'en permettre la vérification.

5.2.2 Prise de décision

Durant son exécution, un agent se fixe un certain nombre de buts, suite à ses observations et à ses interactions avec le monde (perception, communication, négociation). Il se trouve donc confronté au problème de la sélection du but à satisfaire en premier et pour chaque but, de l'action qui permet de l'atteindre.

Face à de telles situations, l'agent analyse les différentes alternatives en termes d'utilité (quel avantage l'agent pourrait en retirer) et d'incertitude (quelle chance a l'action d'être effectuée en fournissant le résultat attendu). Parmi les techniques utilisées pour la résolution de tels conflits, on peut citer l'utilisation de la notion de carte cognitive (*Cognitive Map*) [5] qui est un réseau qualitatif exprimé en terme d'influences (positives ou négatives) et reliant les buts et leur utilité. Le but à retenir sera celui qui présente le plus d'influences positives parmi les buts les plus interdépendants.

La prise de décision est une des caractéristiques des agents rationnels, l'agent tiendra compte de ses croyances pour faire son choix.

5.2.3 Planification

La planification dans les systèmes d'IA classique repose sur l'hypothèse d'un univers statique (seules les actions du planificateur sont prises en compte), cet aspect est incompatible avec l'approche multi-agents [38].

L'IAD, en revanche, offre des possibilités de négociation autorisant une gestion locale des conflits et une planification dynamique. En effet, du fait de l'intervention d'autres agents, un plan peut être remis en cause. L'agent doit pour cela, alterner planification et exécution et réviser des parties de son plan. La planification dans les systèmes multi-agents est une planification distribuée : il n'existe pas de plan global et chaque agent construit son propre plan en coordonnant avec les autres (cas d'agents coopératifs). Certains systèmes d'IAD présentent une planification centralisée, un organe central se chargera de la gestion des conflits et de l'élaboration d'un plan global.

6 Société d'agents

Dans ce paragraphe, nous nous intéressons au comportement global d'une société d'agents et aux théories sociales requises pour la résolution de problèmes dans un univers multi-agents.

6.1 Organisation sociale

L'organisation sociale d'un système multi-agents est la manière dont le groupe est constitué, à un instant donné, pour pouvoir fonctionner. Elle décrit l'ensemble des composants fonctionnels du système, leurs natures, leurs responsabilités et leurs besoins en ressources ainsi que les liens de communication entre les agents. Cette organisation peut être statique ou dynamique.

Selon Gasser [40], une société d'agents est constituée de trois éléments : un ensemble d'agents, un ensemble de tâches à réaliser et un ensemble d'objets associés à l'environnement. Un agent peut prendre la responsabilité d'effectuer une tâche s'il en a la capacité. Il prend alors un rôle dans le groupe. La réalisation d'une tâche suppose la manipulation d'objets de l'environnement.

L'organisation peut être dynamique : le groupe se réorganise à chaque fois en fonction de la tâche à accomplir.

6.1.1 Exemples de structures d'organisation

Les structures organisationnelles en IAD se différencient selon trois facteurs : le type de décentralisation (lié au type de communication), le type des agents appartenant au groupe et le mode de coopération entre les agents.

Cas d'agents réactifs Il n'y a pas vraiment d'organisation dans les systèmes d'agents réactifs. En effet, la structure du système émerge des comportements de ses membres et non d'une volonté d'organisation [75].

Cas d'agents cognitifs Par exemple, les agents du système *Contract Net* (cf. § 2) communiquent selon un protocole de haut niveau et peuvent avoir des rôles différents durant la résolution de problème. Le réseau de communication peut être configuré dynamiquement suivant le nombre des agents, leur localisation et la facilité d'établissement d'une communication.

Bond [12] décrit deux types d'organisation (cf. § 6.3.1) : la structure horizontale et la structure verticale.

Dans le système SYNCHRO [13] [64], on dispose de trois niveaux décisionnels : l'opérateur, le superviseur et l'ensemble des agents (les robots). Le superviseur reçoit les tâches de l'opérateur humain dans un langage relativement abstrait, il les analyse et établit un plan d'actions et une répartition des tâches entre les différents robots. Le système n'est pas complètement décentralisé et les robots possèdent une certaine autonomie décisionnelle leur permettant de contrôler leurs trajectoires, de prendre en compte les obstacles rencontrés et de rendre compte au superviseur du déroulement de leurs missions.

6.1.2 Conclusion

Les travaux sociologiques sur les organisations tels que les modèles développés sur les organisations humaines [51] intéressent la communauté IAD. Dans [62] on distingue cinq types de configurations organisationnelles, elles dépendent du type de décentralisation, du type principal de coordination et

13. SYstème Naturel de Communication Homme-Robot-Ordinateur est un système multi-robots destiné à améliorer la flexibilité des transports dans une unité de production industrielle (la transitique) et peut s'étendre à d'autres applications de la robotique mobile.

des éléments principaux de l'organisation (individus ou autres). Cela va de la structure simple (hiérarchie pyramidale totalement centralisée) à la bureaucratie professionnelle (décentralisation verticale et horizontale) en passant par l'"adhocratie" (fonctionnement par ajustement mutuel entre des petits groupes d'une organisation). Les chercheurs en IAD essayent d'adapter les modèles sociologiques aux systèmes multi-agents.

Dans [76], on propose d'utiliser une bibliothèque de modèles d'organisation. Le choix d'une structure organisationnelle dépend à la fois des caractéristiques du groupe et de la tâche à accomplir, une tâche étant caractérisée par la complexité du raisonnement, la quantité de connaissances et la capacité d'action. Le passage d'une structure à une autre n'est pas sans difficulté et peut échouer. Quelques travaux commencent à s'intéresser aux mesures de l'efficacité d'un système en fonction de la complexité de son organisation sociale [14].

6.2 Contrôle et prise de décision

Dans les systèmes multi-agents, le contrôle peut être exclusivement le résultat des mécanismes de contrôle locaux aux agents alors que dans les sociétés hiérarchisées, certains "agents" (*superviseur*) peuvent prendre en main le contrôle de quelques éléments de la société, en entretenant soit des rapports *maître-esclave* avec les autres agents, soit en guidant la mise en action ou en attente des agents.

Les mécanismes de décision font référence à tout ce qui est allocation de tâches ou de ressources, en relation avec la résolution de conflit. Ils regroupent, entre autres, toute la partie décisionnelle des protocoles de négociation. Les agents entrent en négociation pour résoudre les conflits éventuels (un exemple de protocole de négociation est présenté dans le § 6.4.2).

6.3 Coopération

La coopération [28] [17] est une caractéristique très importante dans les systèmes multi-agents. En effet, une résolution distribuée d'un problème est le résultat de l'interaction coopérative entre les différents agents.

Dans le cadre d'une telle dynamique collective, un agent doit disposer –en plus de la connaissance reflétant son degré d'implication dans cette dynamique (croyances, buts, intentions, engagements, modèle de soi et d'autrui)– d'un certain nombre de compétences nécessaires pour la coopération. Il doit pouvoir :

- mettre à jour le modèle du monde environnant,
- intégrer des informations venant d'autres agents,
- interrompre un plan pour aider d'autres agents,

- déléguer la tâche qu'il ne sait pas résoudre à un autre agent dont il connaît les compétences.

Ces caractéristiques forment les qualités essentielles d'un agent coopératif.

6.3.1 Coopération et structure d'organisation

Selon Bond [12], il existe deux architectures d'organisation pour les sociétés d'agents :

- *Structure horizontale* : Dans de telles sociétés, tous les agents sont au même niveau, il n'y a pas d'agents maîtres et d'agents esclaves. C'est le cas, par exemple, d'un groupe d'agents ayant des spécialités différentes travaillant pour la résolution d'un même problème.

- *Structure verticale* : Dans une architecture verticale, les agents sont structurés par niveaux [14]. Dans un même niveau on retrouve localement une structure horizontale. Le fonctionnement des agents dans de telles sociétés est le suivant : l'agent reçoit le problème à résoudre d'un autre agent qui lui est supérieur dans la hiérarchie, il le décompose en :

  - des sous-problèmes auxquels il peut répondre localement,
  - des sous-problèmes qu'il pourrait résoudre en coopérant avec les autres agents du même niveau que lui,
  - des sous-problèmes qu'il fait suivre aux agents du niveau inférieur dans la hiérarchie.

6.3.2 Modèles de coopération

Quelle que soit l'organisation d'une société d'agents, un agent peut coopérer suivant les modèles suivants [78] :

- Coopération par partage de tâches et de résultats, avec la possibilité de prendre en compte localement les plans des autres.

- Commande : un agent supérieur A décompose le problème en sous-problèmes qu'il répartit entre les autres agents X_i. Ceux-ci le résolvent et renvoient les solutions partielles à A.

- Appel d'offre : A décompose le problème en des sous-problèmes dont il diffuse la liste. Chaque agent X_i qui le souhaite envoie une offre ; A choisit parmi celles-ci et distribue les sous-problèmes. Le système travaille ensuite en mode commande.

14. Agents traitant différents niveaux d'abstraction du problème.

– Compétition : Dans le mode compétition, A décompose et diffuse la liste des sous-problèmes comme dans le mode appel d'offre, chaque agent X_i résout un ou plusieurs sous-problèmes et envoie les résultats correspondants à A qui à son tour fait le tri.

6.4 Résolution de conflits

Les agents coopératifs ont besoin d'éviter autant que possible les situations conflictuelles pour résoudre un problème. Pour ce faire, ils peuvent être amenés à coordonner leurs activités et négocier leurs actions pour arriver à une solution.

6.4.1 Coordination

Les agents travaillent sur des problèmes dont les solutions sont utiles pour les autres agents. Leur travail doit donc être temporellement coordonné. La coordination [27] permet aux agents de considérer toutes les tâches (aucune tâche n'est ignorée) et de ne pas dupliquer le travail. La coordination des actions est liée à la planification et à la résolution des conflits, car c'est à ce niveau qu'on tient compte des actions (plans) des différents agents (cf. § 5.2.3).

Dans les systèmes d'IAD, la coordination des actions des agents peut s'organiser suivant deux schémas principaux [35] : une coordination au moyen d'un système capable de déterminer et de planifier (globalement) les actions des différents agents, ou à l'inverse, on décide de donner une totale autonomie aux agents qui, à leur tour, identifient les conflits pour les résoudre localement.

On peut distinguer deux types de coordination : la coordination due à la gêne (problème de navigation : les agents doivent coordonner leur plans de navigation pour s'éviter mutuellement) et la coordination due à l'aide (la manutention : dans un environnement multi-robots, les agents doivent synchroniser leurs actions pour pouvoir agir efficacement et transporter un objet [64]).

6.4.2 Négociation

Les activités des agents dans un système distribué sont souvent interdépendantes et entraînent des conflits. Pour les résoudre, il faut considérer les points de vue des agents, les négocier et utiliser des mécanismes de décision concernant les buts sur lesquels le système doit se focaliser [20]. La négociation est caractérisée par :

– un nombre faible d'agents impliqués dans le processus ;

– un protocole minimal d'actions : *proposer*, *évaluer*, *modifier* et *accepter* ou *refuser* une solution.

Le processus de négociation ne consiste pas forcément à trouver un compromis mais peut s'étendre à la modification des croyances d'autres agents pour faire prévaloir un point de vue.

Pour mener à bien le processus de négociation [23] [21], il est nécessaire de suivre un protocole qui facilite la convergence vers une solution. Voici un exemple de structure de négociation entre deux agents A et B :

1. A fait une proposition ;
2. B évalue cette proposition et détermine la satisfaction qui en résulte ;
3. Si B est satisfait alors arrêt, sinon B élabore une contre-proposition et donne des arguments ;
4. Si A considère les arguments de B alors reprendre ce schéma en interchangeant A et B, sinon faire intervenir un troisième agent suivant le principe de la *dynamique des échanges* proposé dans [20].

   La dynamique des échanges est à rapprocher de la notion d'animation de réunion. Cette activité effectuée par un agent consiste à ordonner les interventions des agents et tenir compte des différents points de vues, de manière à ce que le groupe aboutisse à une décision.

6.5 Communication

La communication est la base de la résolution coopérative des problèmes. Elle permet de synchroniser les actions des agents et résoudre les conflits de ressources et de buts par la négociation. Dans les systèmes multi-agents, les agents ne disposent d'aucune mémoire commune. La communication entre les agents repose explicitement sur des mécanismes d'envoi de message, de réception et de synchronisation [79].

6.5.1 Protocoles de communication

Dans un système distribué, il faut que les entités disposent d'un langage commun pour pouvoir échanger des informations et coopérer pour la résolution d'un problème. Ce langage appelé mécanisme de communication interprocessus est un ensemble de primitives connues par chaque entité et dont l'ensemble constitue un protocole de communication.

Les primitives de base d'un protocole de communication sont les suivantes :

- établissement d'une connexion entre deux entités ;

- identification du nœud destinataire dans un réseau de communication ;
- envoi de données ;
- réception de données ;
- définition d'un type de communication : *synchrone* ou *asynchrone.*

Les protocoles de communication sont les règles nécessaires à la communication, ils permettent de structurer et d'uniformiser les interactions inter-agents [3]. Un protocole de communication peut être basé sur une architecture standard de communication, en l'occurrence, INTERNET ou OSI. Un exemple de système de création et d'implantation de protocole de communication est présenté dans [65].

6.5.2 Architectures de communication

La communication entre les robots peut être organisée suivant trois schémas différents :

Réseaux en anneau (cf. figure 3) Les interactions dans ce genre d'organisation sont trop lentes, un message destiné à un robot doit transiter par *n-1* robots.

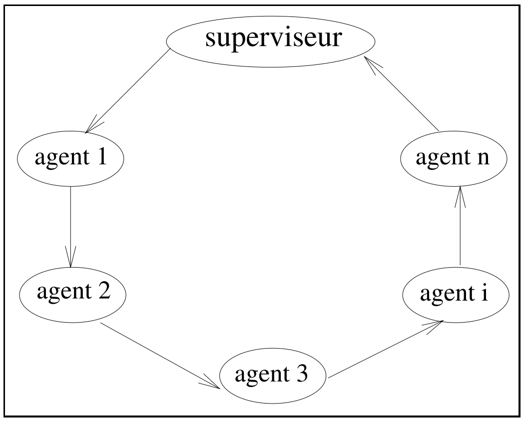

FIG. 3 - réseau en anneau

Réseaux en étoile (cf. figure 4) Ils présentent l'avantage de l'accès rapide entre le superviseur et les autres agents. La nature bidirectionnelle des connexions rend complexe la gestion des interactions inter-agents.

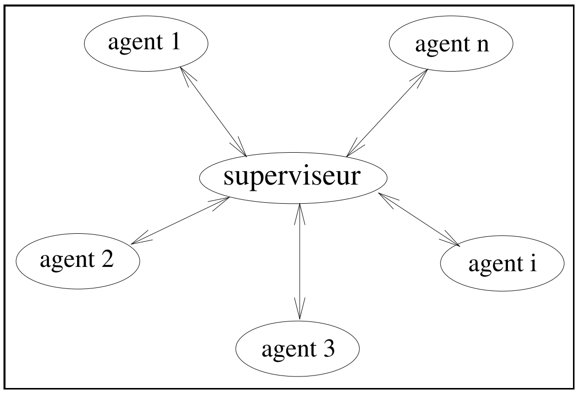

FIG. 4 - réseau en étoile

Réseaux hybrides (cf. figure 5) Ce type d'organisation est une solution hybride reliant deux types d'organisation : un réseau en bus et un réseau en étoile.

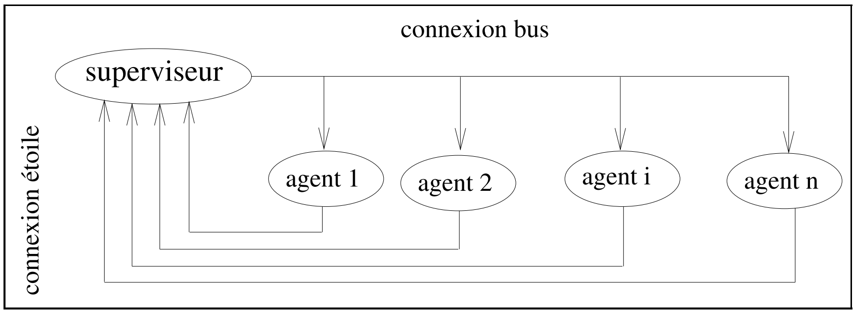

FIG. 5 - réseau hybride

6.5.3 Actes de communication

Une communication dans les systèmes multi-agents n'est pas une simple tâche d'entrée/sortie mais doit être modélisée comme un acte pouvant modifier l'état interne de l'agent récepteur. L'agent peut être amené à remettre en cause son plan d'action, ou à modifier ses croyances.

Une communication peut correspondre à une information, une requête ou une interrogation ; elle doit avoir une sémantique reconnue par l'agent récepteur [15]. La réception d'un message par un agent doit aider à une convergence vers la solution. Tout cela demande une bonne étude (par l'agent émetteur) du contenu du message lors de sa composition.

15. Par exemple, comment décrire un objet de l'environnement à un agent ? Il y a un problème de référence : les objets n'ont pas nécessairement un nom unique ou les mêmes caractéristiques significatives pour tous les agents.

Dans [17], il y a une distinction entre les différents types de communication :

- communication sélective ou diffusée : par opposition à la diffusion, une communication sélective s'adresse à un nombre restreint d'agents, elle suppose donc un critère de sélection ;
- communication sollicitée ou non sollicitée : une communication peut être demandée par un autre agent ;
- communication avec ou sans accusé de réception : dans le cas d'une communication avec accusé de réception, l'émetteur attend une confirmation de la bonne réception du message ;
- communication à transmission simple ou répétée : le message peut être répété sur plusieurs envois.

6.5.4 Modes de communication

Il existe deux principaux modes de communication : la communication par envoi de messages et la communication par partage d'informations.

Communication par envoi de messages Les agents sont en liaison directe et envoient leurs messages directement et explicitement au destinataire. La seule contrainte est la connaissance de l'agent destinataire : "*Si un agent A connaît l'agent B alors il peut entrer en communication avec lui*" [79].

Les systèmes fondés sur la communication par envoi de messages relèvent d'une distribution totale à la fois de la connaissance, des résultats et des méthodes utilisées pour la résolution du problème. Les systèmes d'acteurs en sont l'illustration parfaite (cf. § 3).

Communication par partage d'informations Les composants ne sont pas en liaison directe mais communiquent via une structure de données partagée, où on trouve les connaissances relatives à la résolution (état courant du problème) qui évolue durant le processus d'exécution. Cette manière de communiquer est l'une des plus utilisées dans la conception des systèmes multi-experts. L'exemple parfait d'utilisation de ce mode de communication est l'architecture de *blackboard*, on parle plutôt de sources de connaissances que d'agents. Ce mode de communication n'existe pas dans les systèmes multi-agents où l'on dispose que d'une vision partielle du système alors que la communication par partage d'informations suppose l'existence d'une base partagée sur laquelle les composants viennent lire et écrire.

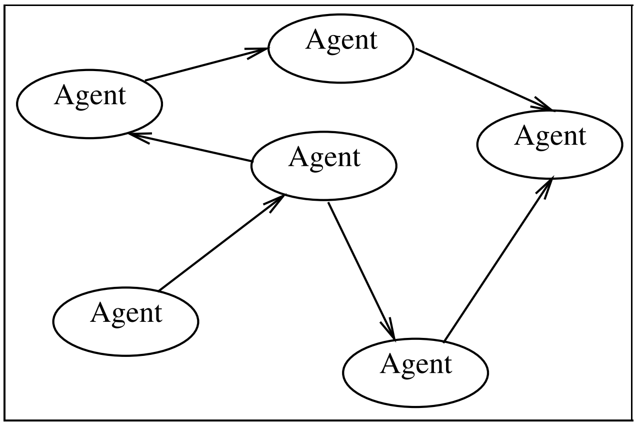

FIG. 6 - communication par envoi de messages

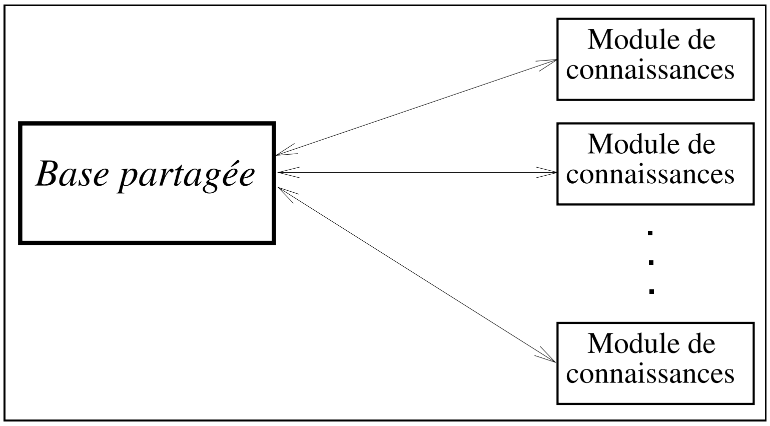

FIG. 7 - communication par partage d'informations

6.5.5 Communication et actes de langages

La théorie des "actes de langages" ou *Speech Acts* [68], est introduite par le philosophe anglais J.-L. Austin dans son livre "*How to do things with words*" [16] [4]. Elle constitue un fondement théorique de la communication basé sur la constatation suivante : lorsqu'on parle, on effectue des actes. Cela apparaît clairement dans le cas des verbes appelés "*performatifs*" (comme promettre, ordonner, demander, etc.) où l'acte est effectué par l'énoncé même : *"je t'ordonne de sortir"*. Mais on peut généraliser aux autres verbes appelés "*constatifs*" (par exemple : "*il pleut*" devient "*je déclare qu'il pleut*").

Austin distingue trois types d'actes dans un énoncé :

- *Actes locutoires* : activité mentale et physique de formulation et d'articulation de l'énoncé. L'acte réussit si l'énoncé est formulé et articulé correctement ;

- *Actes illocutoires* : actes effectués par l'interlocuteur lors de la formulation de l'énoncé. En énonçant "*il est dix heures ?*", on peut comprendre s'il s'agit d'une affirmation ou d'une interrogation. Faire comprendre qu'il s'agit bien d'une interrogation est un acte illocutoire qui réussit. L'acte illocutoire est donc codé dans le message. Dans cet exemple, si l'auditeur comprend que c'est une affirmation, alors l'acte illocutoire échoue ;

- *Actes perlocutoires* : c'est la conséquence indirecte mais pouvant être intentionnelle des actes locutoires et illocutoires sur l'auditeur. Ces actes dépendent du contexte, et contrairement aux actes illocutoires, ils ne sont pas codés dans l'énoncé. Par exemple à l'énoncé *"il fait froid"* est associé l'acte perlocutoire de mise en marche du chauffage.

Ce sont surtout les actes illocutoires qui ont fait l'objet de nombreux travaux dont notamment ceux de J. Searle [68] qui a proposé une nouvelle taxinomie d'actes illocutoires.

La théorie des actes de langages [74] a constitué un premier fondement théorique pour la spécification de modules de communication. Son intérêt provient surtout de l'uniformité des mécanismes qu'elle propose et de son accessibilité à des non philosophes, ce qui a permis de l'intégrer dans d'autres disciplines.

En IA, l'approche symbolique et les travaux déjà existants sur la modélisation de l'action [44] [2] étaient un atout favorable pour l'introduction de cette théorie en 1979 par P. Cohen et C. Perrault du MIT. Parmi ses applications, la réalisation d'interfaces homme-machine en langage naturel [39] [56], la réalisation de systèmes d'aide à l'organisation [80] et l'enseignement intelligemment assisté par ordinateur [19].

16. Traduit en français en "Quand dire, c'est faire".

Dans les travaux sur les systèmes multi-agents, plusieurs chercheurs sur la communication commencent à s'intéresser à la théorie des actes de langages [71] [15] [13]. Cette théorie permet une bonne modélisation de la communication entre les agents. Les actes de langages permettent de bien illustrer la théorie des intentions, en effet, à une intention correspond un certain type d'actions (actes illocutoires). L'intention, et par conséquent la communication, seront donc reconnues à travers le type de l'action (déclaration, interrogation, requête, offre, etc.). Cela impose une mise en correspondance entre les intentions et les modèles internes de l'émetteur et du récepteur des messages.

En décomposant ainsi les dialogues en "actes de langages", les structures de communication et le phénomène "perception/action" deviennent homogènes.

Toutefois, la théorie des actes de langages reste insuffisante pour la modélisation de la coordination et de la négociation et donc de la coopération entre les agents car elle ne permet pas de décrire les informations échangées, quand ont-elles été échangées, quel agent fait quoi, la décomposition et l'allocation des tâches, etc.

6.6 Émergence

L'émergence est un concept relatif aux systèmes réactifs. C'est un phénomène en cours d'exploration dans les SMA réactifs, il est d'autre part étudié en physique, en biologie, et en systémique.

Une fonctionnalité qui émerge est une fonctionnalité qui n'a pas été explicitement programmée, elle est le résultat d'interaction entre deux ou plusieurs comportements. En effet, la fonctionnalité accomplie par une société n'est pas la simple somme des fonctionnalités de ses composants, mais résulte de l'action commune des individus en interdépendance. Cette notion ressemble beaucoup à l'éco-résolution [17] [36].

Par exemple, si on veut qu'un robot (en marche) suive un mur (comportement *suivre le mur*), on définit deux comportements (antagonistes) : *être attiré par le mur* et *être repoussé par le mur*. De l'interaction de ces deux comportements va émerger un nouveau comportement : *suivre le mur*.

Dans ce cas, on parle d'émergence de comportement, mais il existe aussi d'autres formes d'émergence telles que l'émergence de rôles, l'émergence de stratégies, l'émergence de contrôle et l'émergence des structures de coordina-

17. L'éco-résolution est une technique de résolution de problème par coopération entre plusieurs entités. La résolution s'effectue par la dynamique d'évolution des différentes entités qui adaptent leurs comportements locaux aux contraintes de l'environnement jusqu'à ce que les buts soient atteints. L'éco-résolution conduit à l'éco-programmation (*computational ecology*) [48].

tion. L'émergence est un phénomène compliqué et très peu compris, beaucoup de chercheurs s'y intéressent, notamment en France dans l'équipe de J. Erceau.

6.7 Conclusion

L'approche IAD consiste à élaborer une architecture constituée de composants d'IA (systèmes experts, etc.) indépendants coopérant pour la résolution d'un problème avec une gestion des conflits éventuels assurée généralement par un organe central de contrôle.

Dans les sytèmes multi-agents, le contrôle est totalement distribué, chaque entité appelée agent, possède une vision partielle de son environnement avec lequel elle interagit. Les agents dépassent les simples entités et peuvent manipuler des concepts de haut niveau comme la rationalité, l'intentionnalité, etc.

Les systèmes multi-agents sont donc des systèmes d'IAD qui respectent les règles des systèmes ouverts et dont les entités (agents) peuvent manipuler des concepts de haut niveau.

7 Travaux en IAD

Les recherches en IAD sont très actives et des systèmes opérationnels ont déjà été mis au point. Dans cette section, nous décrivons brièvement quelques travaux et des exemples de systèmes d'IAD développés récemment.

Nous commençons par donner une classification des travaux actuels sur l'IAD que nous répartissons comme suit :

- travaux théoriques sur les systèmes multi-agents : organisation d'une société d'agents, compréhension de phénomènes tels que l'émergence, la résolution de problème dans une société d'agents, etc. ;
- développement d'outils pour l'IAD : langages, méthodologies de conception, plate-forme de développement de systèmes d'IAD ;
- utilisation de l'architecture IAD comme une technologie de développement de systèmes informatiques (industriels et autres).

Les applications de l'IAD ont touché de nombreux domaines comme le diagnostic, l'enseignement, la conception, l'acquisition des connaissances, la reconnaissance des formes, la robotique, la compréhension de langage naturel, le contrôle de réseaux de communication ou d'énergie, etc.

Beaucoup de laboratoires et projets Esprit s'intéressent au thème multi-agents. En France, on peut citer par exemple, le Laboratoire d'Automatique et d'Analyse des Systèmes de Toulouse (LAAS-CNRS) qui adopte une approche multi-agents dans le domaine de la robotique. Le laboratoire d'informatique de Grenoble (LIFIA) s'intéresse à l'IAD pour le développement d'un système de gestion de réseaux de distribution ou de production.

Au niveau européen, plusieurs projets Esprit II sont consacrés à l'IAD. C'est le cas du projet ARCHON (ARchitecture for Cooperating Heterogeneous ON-line systems) qui concerne la conception et la réalisation de prototypes de coopération entre systèmes experts pour la conduite de processus.

Le projet KBS-SHIP concernant la conception d'un système d'aide à la prise de décision quant au suivi du plan de navigation d'un navire, d'assurer la surveillance et le contrôle des machines, de prévenir les avaries, etc.

Citons aussi le projet Esprit IMAGINE[18] destiné à la réalisation d'un atelier de génie logiciel pour les systèmes multi-agents.

Le projet SKIDS (Signal and Knowledge Integration with Decisional control for multisensorial Systems) vise la conception d'une machine de perception multisensorielle temps-réel dédiée à la robotique et à certains systèmes de surveillance.

Dans ce qui suit, nous décrivons brièvement quelques systèmes d'IAD, en l'occurence HOPES, HECODES, ARCHON et SAHARA. On évoquera enfin un exemple de travaux sur les systèmes réactifs.

7.1 HOPES

La structure hiérarchique de HOPES [7] (Hierarchically Organized Parallel Expert System) est bien adaptée à la résolution de problème par niveau. Le cheminement du flot d'informations entre agents est effectué au moyen de *blackboards* partagés par les agents à plusieurs niveaux.

Par exemple, dans la figure 8, l'agent Ax sera responsable de la centralisation des résultats de la coopération entre les agents $Ax_1, ..., Ax_4$ dans le blackboard B. Les résultats sont affichés sur le blackboard pour être utilisés par l'agent Ax, mais B est aussi utilisé par les agents du niveau inférieur qui, à leur tour, interagissent pour la résolution collective d'une tâche. Ils utilisent B comme un canal durant leur communication intra-groupe.

D'autres agents au même niveau que Ax peuvent vouloir utiliser ce résultat ou collaborer avec Ax d'une autre façon. Pour ce faire, ils utiliseraient un blackboard H de plus haut niveau. Cette stratification peut être répétée pour aboutir à des niveaux multiples.

18. Auquel participe la société Française STERIA.

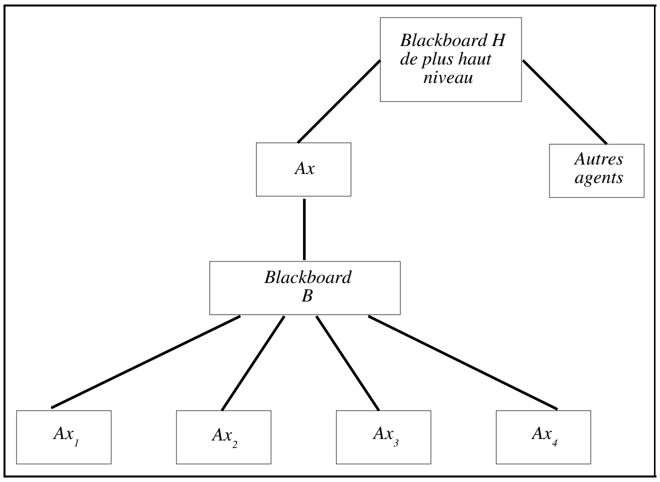

FIG. 8 - modèle hiérarchique de HOPES

7.2 HECODES

HECODES (HEterogeneous COoperating Distributed Expert System) [7] est un environnement approprié pour une coopération horizontale, hiérarchique et récursive. Il est aussi basé sur un système de *blackboard*. Il présente un système de contrôle centralisé, un système de blackboard et des systèmes experts. Les agents coopérant peuvent être hétérogènes par rapport à leur stratégie de contrôle, leur méthode de représentation des connaissances et leur langage de programmation.

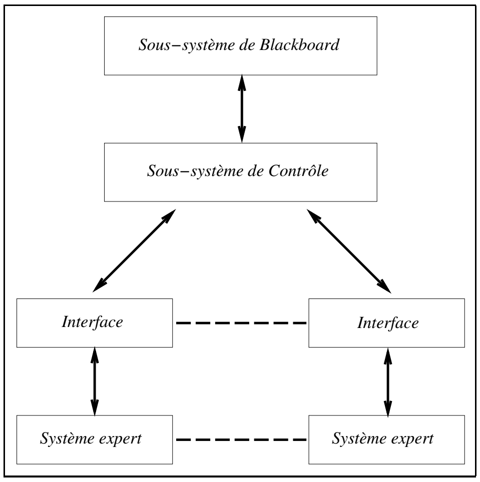

FIG. 9 - architecture de HECODES

Dans la figure 9, les systèmes experts, le sous-système de contrôle et le sous-système de *blackboard* représentent les nœuds d'un réseau reliés par des canaux de télécommunication. Chaque système expert peut fournir des solutions et résoudre des problèmes locaux de façon autonome en utilisant sa propre connaissance du domaine. Toutefois, chacun peut avoir besoin de l'assistance des autres agents ou alors peut la leur fournir à travers le sous-système de contrôle. Il existe des *interfaces* entre les systèmes experts (SE) et le sous-système de contrôle qui servent d'interfaces de communication, d'interfaces homme-machine et de gestion de SE hétérogènes.

Le sous-système de *blackboard* centralise les informations partagées par les systèmes experts. Le sous-système de contrôle est responsable de la gestion de la coopération et de la communication entre les nœuds et du temps d'exécution de HECODES. En pratique, les sous-systèmes de blackboard et de contrôle sont regroupés dans un nœud central pour minimiser les communications.

7.3 ARCHON

ARCHON (ARchitecture for Cooperating Heterogeneous ON-line systems) [66] est une architecture pour le contrôle de processus industriels. Les principales motivations étaient de concevoir une application industrielle (non limitée à un domaine spécifique) pour encapsuler des systèmes experts existants en vue de la coopération. Les éléments intervenant dans un tel système sont fortement hétérogènes et leur niveau d'intelligence est très variable, ce qui nécessite des moyens de communication de différents niveaux. Ce système se compose essentiellement de quatre parties : un module de communication, un module de planification et de coordination, un module de coopération et un module de contrôle pour la gestion dynamique de l'interface du système expert.

La communication dans le système ARCHON est de haut niveau, elle introduit les concepts d'adressage intelligent, de filtrage et d'ordonnancement.

- *L'adressage intelligent* permet à un agent d'envoyer un message à un groupe d'agents intéressés par ce message.
- *Le filtrage* permet à un agent de recevoir uniquement des messages pertinents ou traitant d'un sujet précis. L'adressage intelligent est un filtrage des messages envoyés.
- *L'ordonnancement* permet aux agents de recevoir des messages dans un ordre spécifique. Cet ordre est défini à partir d'une description abstraite du contenu du message ou à partir du type du message.

7.4 SAHARA

SAHARA (Simulateur d'Architectures Hétérogènes d'Agents et de Ressources Actives) [8] est un outil d'optimisation de systèmes multi-experts, basé sur un simulateur d'architecture. Le but de cet outil est d'aider les concepteurs à déterminer la granularité d'un système et à spécifier les différents modules et d'en évaluer les performances. Cette évaluation s'effectue d'une part en observant le comportement des modules et d'autre part, grâce à des mesures sur les temps de traitement, sur l'occupation mémoire ou sur les communications.

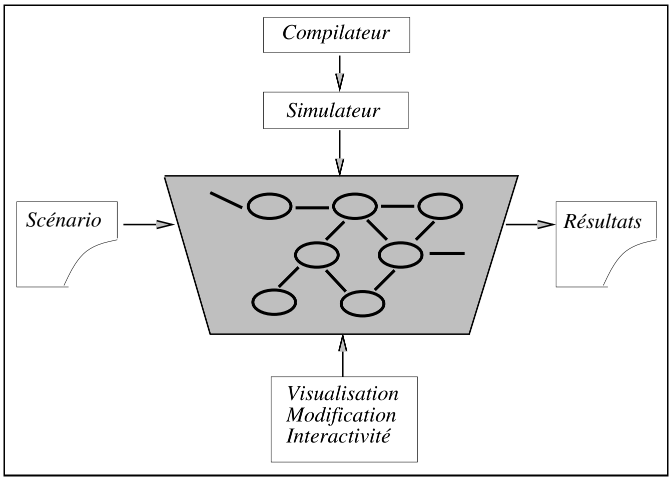

FIG. 10 - architecture de SAHARA

7.5 Application d'agents réactifs

Les systèmes qu'on vient d'évoquer sont tous des systèmes d'IAD d'une façon générale, peu de systèmes multi-agents existent mais un grand intérêt est exprimé par la communauté des chercheurs pour de tels systèmes. Dans ce sens on peut citer les travaux de l'équipe MIRIAD dirigée par le Pr J. Ferber au LAboratoire FORmes et Intelligence Artificielle (LAFORIA, Paris VI) sur les systèmes d'agents réactifs.

Dans les systèmes réactifs, l'intelligence émerge des interactions d'un grand nombre d'agents qui ne disposent d'aucune intelligence. L'analogie classique de telle société d'agents est celle de la fourmilière ou de la ruche d'abeilles. Chaque insecte est relativement simple et ne dispose que d'un ensemble très réduit de comportements. Pourtant, la société d'insectes fonctionne comme si elle disposait d'un organe capable de prendre les décisions et donnant des ordres aux éléments de la société.

Outre la modélisation des sociétés de fourmis, l'équipe MIRIAD travaille sur l'application des systèmes d'agents réactifs au trafic aérien [67], par exemple, afin de coordonner les trajectoires d'un grand nombre d'avions, dans des couloirs aériens, chaque avion étant considéré comme un agent réactif autonome.

8 Conclusion

Dans ce rapport, nous avons présenté une étude des univers multi-agents, nous avons proposé une architecture d'agents intelligents et essayé de clarifier la terminologie du domaine qui est actuellement confuse : IAD, SMA et Systèmes Ouverts.

L'IAD et les SMA sont des thèmes de recherche en cours d'exploration. Ils font intervenir plusieurs domaines de recherche tels que les systèmes répartis, la biologie, l'IA, la psychologie cognitive et la sociologie. De ce fait, les termes utilisés varient selon les différents chercheurs ; il est donc nécessaire dans une première étape de définir un vocabulaire commun compréhensible quel que soit le domaine de recherche évoqué [60] et dans une seconde étape de définir les frontières et les limites des thèmes en question afin d'éviter toute source de confusion avec les systèmes répartis [53].

Les concepteurs d'application d'IAD expriment aussi le besoin d'une architecture de base pour la construction de systèmes multi-agents avec un langage de spécification [52] [72], des protocoles de communication et de coopération [25] ainsi que des moyens de mise en œuvre des différentes lois sociales dans une société d'agents.

Nous avons proposé une synthèse des travaux en IAD et SMA. Les principales conclusions que nous pouvons en tirer sont les suivantes :

- l'approche distribuée de l'IAD s'adapte mieux à la réalité des systèmes complexes que l'IA classique, en effet les comportements que l'on vise à modéliser sont le plus souvent le résultat d'une coopération entre plusieurs entités ;

- l'IAD enrichit le processus de résolution de problèmes en le partageant entre plusieurs agents ;

- l'interaction de plusieurs expertises incomplètes ou peu fiables peut mener à une expertise plus sûre et plus robuste ;

- l'évolution des machines parallèles est un atout favorable à cette approche ;

- l'IAD permet de respecter les normes du génie logiciel dans l'élaboration des systèmes, à savoir : la modularité, la fiabilité et la réutilisabilité ;

- l'IAD est pluridisciplinaire, un système multi-agents doit intégrer certains mécanismes régissant la dynamique de participation d'un individu à la vie du groupe (ce qui nécessite des notions sociologiques, etc. [40]).

Beaucoup de systèmes existent mais la plupart présentent une centralisation du contrôle et n'intègrent pas des concepts comme l'intentionnalité ou la rationalité. On est donc loin des vrais systèmes multi-agents tels qu'ils sont théoriquement définis.

Les systèmes d'acteurs constituent un pas vers une architecture multi-agents, mais les travaux sur les acteurs tiennent plus de la conception de langages que de systèmes de résolution de problèmes.

Il est vrai que l'évolution des recherches sur les SMA dépend de l'avancement des recherches sur la modélisation de phénomènes tels que l'intentionnalité, la rationalité, les engagements et la représentation des croyances qui n'ont pas encore atteint le stade de la concrétisation.

References

[1] G. Agha. Concurrent Object–Oriented Programming. *Communication of the ACM*, 33(9), September 1990.

[2] J. F. Allen. Towards a General Theory of Action and Time. *Artificial Intelligence 23*, pages 123–154, 1984.

[3] D. Appelt. Planning English Referring Expressions. Report, 1980.

[4] J. L. Austin. *How to Do Things with Words.* Oxford University Press, 1962. (Éditions du Seuil, 1970, en français).

[5] R. Axelrod. *Structure of Decision: The cognitive maps of Political Elites.* Ptinceton University Press, 1976.

[6] H. S. Becker. Notes of the Concept of Commitment. *American Journal of Sociology 66*, pages 32–40, July 1960.

[7] D. Bell and J. Grimson. *Distributed Database Systems.* Addison–Wesley, 1992.

[8] P. Benhamou, M. Barat, and J. Ferber. Sahara : Simulateur d'architectures hétérogènes d'agents et de ressources actives. In 10^{eme} *Journées Internationales d'Avignon EC2*, volume 1, pages 793–805, mai 1990.

[9] A. Bond and L. Gasser. *An Analysis of Problems and Research in DAI*, chapter 1. Morgan Kaufmann, California, 1988.

[10] A. H. Bond. Commitment: A Computational Model for Organizations of Cooperating Intelligent Agents. In *Proceedings of the 1990 Conference on Office Information Systems*, pages 21–30, Cambridge, 1990.

[11] A. H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence.* Morgan Kaufmann, Palo Alto, CA, 1988.

[12] A.H. Bond. Distributed Decision Making in Organisation. *IEEE Transactions on Systems, Man and Cybernitics Conference*, November 1990.

[13] T. Bouron. COMMAS: A COmmunication MOdel for MUlti–Agent Systems. In *European Simulation Muticonferences: ESM'91*, pages 220–225, 1991.

[14] T. Bouron. Conception d'agents sociaux et évaluation de leurs performances. In *Actes des* 3^{emes} *journées "Symbolique-numérique"*, mai 1992.

[15] T. Bouron. *Structure de Communication et d'Organisation pour la Coopération dans un Univers Multi-agents: Une contribution à la définition d'un moèle de programmation agent basé sur les concepts d'engagement et d'acte de langage.* PhD thesis, Université Pierre et Marie Curie (Paris VI), Novembre 1992.

[16] R. A. Brooks. A Robot that Walks: Emergent Behaviors From a Carefully Evolved Network. A.I. Memo 1091, Massachusetts Institute of Technology, 1989.

[17] S. Cammarata, D. McArthur, and R. Steeb. Strategies of Cooperation in Distributed Problem Solving. In Karlsruhe, editor, *Proceedings of the* 8^{th} *Internatinal Joint Conference on Artificial Intelligence*, volume 2, pages 767–770, Germany, August 1983.

[18] B. Chaib-draa and E. Paquet. Architecture d'un Système Intelligent pour les environnements multi-agents. In *Actes des journées Volcans-IA 93*, Clermont-Ferrand, Mars 1993.

[19] R. Chevallier. STUDIA*: Mise en œuvre d'un modèle dynamique de dialogue dans un tuteur intelligent.* PhD thesis, Université de Maine, 1992.

[20] V. Chevrier. Coordination et structuration des échanges par négociation dans les systèmes multi–agents. In *Journée Systèmes Multi–Agents PRC–GDR Intelligence Artificielle*, Nancy, décembre 1992.

[21] S. E. Conry, R. A. Meyer, and V. R. Lesser. *Multi–Stage Negotiation in Distributed Planning*, pages 367–384. Morgan Kaufmann, California, 1988.

[22] R. Davis. Report on the Workshop on Distributed AI. *SIGART Newsletter 73*, pages 42–52, 1980.

[23] R. Davis and R. G. Smith. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence*, 20:63–109, 1983.

[24] R. Dieng. Relations Linking Cooperating Agents. In *Proceedings of the* 2^{nd} *European Workshop MAAMAW'90*, pages 185–202, Saint-Quentin en Yvelines-France, August 1990.

[25] C. Djossou. A Client–Server Based Architecture for Communication Between Expert Systems. In W.W. Koczkodaj, P.E. Lauer, and A.A. Toptsis, editors, *4th International Conference on Computing and Information*, pages 231–235, Canada, May 1992.

[26] E. H. Durfee, V. R. Gasser, and D. D. Korkill. Coherent Cooperation Among Communicating Problem Solvers. *IEEE Transactions on Computers C–36*, pages 1275–1291, 1987.

[27] E. H. Durfee and V. R. Lesser. Using Partial Global Plans to Coordinate Distibuted Problem Solving. In *Proceedings of the* 10^{th} *IJCAI*, pages 875–883, Milan, Italy, 1987.

[28] E. H. Durfee, V. R. Lesser, and D. D. Corkill. *Cooperative Distributed Problem Solving*, volume 4, pages 83–147. Addison Wesley, 1989.

[29] E. H. Durfee and T. A. Montgomery. A Hierarchical Protocol for Coordinating Multiagent Behaviors. In *Proceedings of the* 8^{th} *National Conference on Artificial Intelligence*, pages 86–93, Cambridge, July–August 1990. Volume One.

[30] J. Erceau and J. Ferber. L'intelligence artificielle distribuée. *La Recherche*, 22:750–758, juin 1991.

[31] L. Erman, F. Hayes-Roth, V. Lesser, and R. Reddy. The HEARSAY-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty. *Computing Surveys*, 12:213–253, 1980.

[32] M. Fehling and L. Erman. Report on the Third Annual Workshop on DAI. *SIGART Newsletter 84*, pages 3–12, April 1983.

[33] J. Ferber. *MERING: un langage d'acteur pour la représentation et la manipulation des connaissances.* PhD thesis, Université Pierre et Marie Curie (Paris VI), 1983.

[34] J. Ferber. Mering: An Open-Ended object oriented language for knowledge representation. In *ECAI'84*, September 1984.

[35] J. Ferber. *Conception et Programmation par Objets.* Technologies de pointe - informatique, Hermes, 1990.

[36] J. Ferber. The Framework of Eco-Problem Solving. In *Proceedings of the* 2^{nd} *European Workshop MAAMAW'90*, pages 103–114, Saint-Quentin en Yvelines-France, August 1990.

[37] J. Ferber and J.-P. Briot. Design of a Concurrent Language for Distribueted Artificial Iintelligence. Technical report, Université de Paris 6, 88/21 1988.

[38] J. Ferber and M. Ghallab. Problématiques des univers multi–agents intelligents. In *Actes des journées nationales PRC–GRECO Intelligence Artificielle*, pages 295–320, Toulouse, mars 1988.

[39] J. R. Galliers. *A Theorical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi–Agent Conflict.* PhD thesis, Open University (UK), 1988.

[40] L. Gasser. Social Conceptions of Knowledge and Action. Technical Report ACT-AI-355-90, MCC, October 1990.

[41] L. Gasser, C. Braganza, and N. Herman. MACE: A Flexible Testbed for Distributed AI Research. Distributed Artificial Intelligence vol. 1, 1987.

[42] M. L. Ginsberg. *Decision Procedures*, pages 3–28. Morgan Kaufmann, 1987.

[43] J. Y. Halpern and Y. Moses. Knowledge and Common Knowledge in a Distributed Environment. In *3rd ACM Conference Principles of Distributed Computing*, pages 50–61, 1984.

[44] B. Hayes-Roth. A Blackboard Model of Control. Technical Report HPP83-38, Stanford University, 1984.

[45] C. Hewitt, P. Bishop, and R. Steiger. A Universal Modular Actor Formalism for Artificial Intelligence. In *Proceedings of the the* 3^{rd} *IJCAI*, pages 235–245, Standford, California, 1973.

[46] C. E. Hewitt. Viewing Control Structure as Patterns of Passing Messages. *Artificial Intelligence 8*, pages 323–364, 1977.

[47] C. E. Hewitt and H. Lieberman. Design Issues in Parallel Architectures for Artificial Intelligence. A.I. Memo 750, Massachusetts Institute of Technology, 1983.

[48] B. Huberman. *The Ecology of Computation.* Elsevier Science, North Holland, 1988.

[49] M. N. Huhns. Distributed Artificial Intelligence. Pitman Publishing–Morgan Kaufman, 1987.

[50] K. Konolige. Agents with Attitudes. Invited talk/panel at the AAAI'91, SRI International, 1991.

[51] W. Kornfeld and C. Hewitt. The scientific community metaphor. *IEEE Transactions on Systems, Man and Cybernitics Conference*, SMC–11(1), 1981.

[52] S. Labidi. Réseaux sémantiques d'agents coopérants. Technical report, Université de Nice, septembre 1991. Rapport de DEA.

[53] W. Lejouad. Un générateur de systèmes experts réparti est-il un système multi–agents? In *Journée Systèmes Multi–Agents PRC–GDR Intelligence Artificielle*, Nancy, décembre 1992.

[54] V. Lesser and D. Corkill. The Distributed Vehicle Monitoring Testbed: a Tool for Investigating Distributed Problem Solving Networks. *AI Magazine*, pages 15–33, 1983.

[55] V. R. Lesser and D. D. Corkill. *Distributed Problem Solving Networks*, pages 245–251. John Wiley and Sons, New York, 1987.

[56] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On Acting Together. In *Proceedings of the 8^{th} National Conference on Artificial Intelligence*, pages 94–99, Cambridge, July–August 1990.

[57] H. Lieberman. *Concurrent Object–Oriented Programming in Act1*, pages 9–36. MIT Press, 1987.

[58] G. Masini, A. Napoli, D. Colnet, D. Léonard, and K. Tombre. *Les langages à objets : langages de classes, langages de frames, langages d'acteurs.* InterEditions, Paris, 1989.

[59] Equipe Miriad. Approcher la notion de collectif. In *Journée Systèmes Multi–Agents PRC–GDR Intelligence Artificielle*, Nancy, décembre 1992.

[60] Pôle et lieu d'échange en IAD. Vers une taxinomie du vocabulaire pour les systèmes multi–agents. In *Journée Systèmes Multi–Agents PRC–GDR Intelligence Artificielle*, Nancy, décembre 1992.

[61] M. Minsky. *The Society of Mind.* Basic Books, 1986. (la société de l'esprit, Inter Editions, 1988, en français).

[62] H. Mintzberg. Structure et dynamique des organisations. Les éditions d'organisation (Paris) et les éditions Agence d'ARC Inc. (Montreal), 1986.

[63] A. Newell. The Knowledge Level. *Artificial Intelligence 18*, pages 87–127, 1982.

[64] C. Pegard. *Coordination de robots mobiles Autonomes, application aux chantiers automatisés.* PhD thesis, Université de Picardie, 1988.

[65] P. Populaire, O. Boissier, and J. Sichman. Description et Implémentation de Protocoles de Communication en Univers Multi-agents. In *Actes des* 1^{eres} *journées Francophone IAD et SMA*, pages 241–252, Toulouse, Avril 1992.

[66] C. Roda, N. R. Jennings, and E. H. Mamdani. ARCHON: A Cooperation Framework for Industrial Process Control. In *Proceedings of the Int. Working Conference on Cooperating Knowledge Based Systems*, pages 95–112. Springer–Verlag, october 1990.

[67] A. Sari. Informatique et hémopoïèse. *Dossier la recherche scientifique et médicale*, pages 12–13, 1992.

[68] J. R. Searle. *Speech Acts.* Cambridge University Press, 1969. Herman 1972, (en français).

[69] J. R. Searle. *Intentionality: An Essay in the philosophy of mind.* Cambridge University Press, New York, 1983.

[70] J. R. Searle. *Intentions in Communication*, chapter 19:Collective Intentions and Actions, pages 401–415. MIT Press, London, 1990.

[71] Y. Shoham. Agent–Oriented Programming. Technical Report Stan-CS-1335-90, Stanford University, 1990.

[72] Y. Shoham. AGENT0: A simple agent language and its interpreter. In AAAI Press/The MIT Press, editor, *Proceeding of the* 9^{th} *National Conference on Artificial Intelligence, AAAI-91*, pages 704–709, July 1991.

[73] R. G. Smith. *The Contract Net Protocol: High–Level Communication and Control in a Distributed Problem Solver*, pages 357–366. Morgan Kaufmann, California, 1988.

[74] D. Sperber and D. Wilson. Relevance Communication and Cognition. Éditions de minuit en français, 1986.

[75] L. Steels. Cooperation between distributed agents through self-organisation. In *Decentralized artificial intelligence*, volume 1. Y. Demazeau and Muller, 1990.

[76] E. Strugeon, R. Mandiau, and G. Libert. Proposition d'organisation dynamique d'un groupe d'agents en fonction de la tâche. In *Actes des* 1^{eres} *journées Francophone IAD et SMA*, pages 217–227, Toulouse, Avril 1992.

[77] Tuomela, Raimo, Kaarl, and Miller. We intentions. *Philosophical Studies*, 53:367–389, 1988.

[78] A. Vailly and M-A. Simon. Des systèmes experts coopérants, pourquoi, comment? In *Cognitiva 87* AFCET, pages 183–188, Paris, 1987.

[79] F. Vernadat and P. Azéma. Prototypage de systèmes d'agents communicants. In *Journée Systèmes Multi–Agents PRC–GDR Intelligence Artificielle*, Nancy, décembre 1992.

[80] T. Winograd. Language/Action Perspective On the Design of Cooperative Work. *Human–Computer Interaction*, 3(1):3–30, 1987.

[81] A. Yonezawa, J.P. Briot, and E. Shibayama. Object–Oriented Programming in ABCL/1. In 1^{st} *OOPSLA*, pages 258–268, Oregon, 1986.

[82] A. Yonezawa, E. Shibayama, T. Takada, and Y. Honda. *Modelling Programming in an Object–Oriented Concurrent Language ABCL/1*, pages 55–89. MIT Press, 1987.

List of Figures

Contents

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur

INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)

ISSN 0249-6399