

Systèmes multiagents : Principes généraux et applications

B. Chaib-draa, I. Jarras et B. Moulin

Département d'Informatique, Pavillon Pouliot,
Université Laval, Ste-Foy, PQ, Canada, G1K 7P4

Téléphone : (418) 656-2131 poste 3226

Télécopieur : (418) 656-2324

adr-élec : {chaib, moulin, jarras}@ift.ulaval.ca

Résumé

Le thème des systèmes multiagents (SMAs), s'il n'est pas récent, est actuellement un champ de recherche très actif. Cette discipline est à la connexion de plusieurs domaines en particulier de l'intelligence artificielle, des systèmes informatique distribués et du génie logiciel. C'est une discipline qui s'intéresse aux comportements collectifs produits par les interactions de plusieurs entités autonomes et flexibles appelées agents, que ces interactions tournent autour de la coopération, de la concurrence ou de la coexistence entre ces agents. Ce chapitre introduit, tout d'abord, les notions d'agents et de systèmes multiagents (SMAs), et détaille par la suite les différentes questions que soulèvent la problématique des SMAs, en particulier : les interactions et la coopération, la coordination, la planification et la communication. À la fin, un aperçu des différentes applications des SMAs est donné.

1. Des agents autonomes aux systèmes multiagents

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

1.1 Qu'est ce qu'un agent?

Le concept d'agent a été l'objet d'études pour plusieurs décennies dans différentes disciplines. Il a été non seulement utilisé dans les systèmes à base de connaissances, la robotique, le langage naturel et d'autres domaines de l'intelligence artificielle, mais aussi dans des disciplines comme la philosophie et la psychologie. Aujourd'hui, avec l'avènement de nouvelles technologies et l'expansion de l'Internet, ce concept est encore associé à plusieurs nouvelles applications comme *agent ressource*, *agent courtier*, *assistant personnel*, *agent interface*, *agent ontologique*, etc. Dans la littérature, on trouve une multitude de définitions d'agents. Elles se ressemblent toutes, mais diffèrent selon le type d'application pour laquelle est conçu l'agent. Nous trouvons dans [55] une discussion sur les différentes définitions attribués aux agents ainsi que la différence entre un agent et un programme classique. À titre d'exemple, voici l'une des premières définitions de l'agent dûe à Ferber [53] :

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

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

Récemment, Jennings, Sycara et Wooldridge [74] ont proposé la définition suivante pour un agent :

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

Les notions "situé", "autonomie" et "flexible" sont définies comme suit :

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

Cette définition est conforme à notre vision de l'entité agent et par conséquent nous avons convenu de l'adopter tout au long de ce chapitre. De ce fait, nous nous référons donc à des agents cognitifs capables de planifier et de prendre des décisions et non à des agents réactifs comme le verrait l'écologie [19].

Bien entendu, dépendamment des applications certaines propriétés sont plus importantes que d'autres, il peut même s'avérer que pour certaines types d'applications, des propriétés additionnelles soient requises. Il convient cependant de souligner que la présence des propriétés qu'on vient de voir comme l'autonomie, la flexibilité, la sociabilité, etc., donne naissance au paradigme agent tout en le distinguant des systèmes conventionnels comme les systèmes distribués, les systèmes orientés objets et les systèmes experts.

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

1.1.1 Historique

La majeure partie des composantes d'un agent autonome provient de l'intelligence artificielle. On serait donc porté à croire que la construction d'un agent autonome serait au centre des problèmes étudiés par l'intelligence artificielle (IA). Ce n'est cependant pas le cas. En effet, avant les années 80, la recherche en intelligence artificielle s'est plutôt concentrée sur les diverses composantes d'un agent, en supposant qu'on pourrait un jour assembler ces différentes parties afin de construire un agent de manière assez directe. Durant cette période, le champ d'intérêt qui a le plus en commun avec les agents autonome est celui de la planification.

La planification en IA : La planification [3,29,104,145] est le sous-domaine de l'intelligence artificielle qui cherche à répondre à la question: "Que doit-on faire?", c'est-à-dire quelle action poser et dans quel ordre. Comme un agent est une entité qui pose des actions, on voit rapidement l'intérêt de la planification pour les agents autonomes. Un des premiers systèmes de planification à être développé fut le système STRIPS (STanford Research Institute Problem Solver, la distinction entre planificateur et solutionneur n'étant pas encore claire à l'époque) [55]. Dans un système STRIPS typique, on retrouve habituellement les composantes suivantes :

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

Dans les années 70 jusqu'au début des années 80, la recherche sur la planification s'est donc surtout orientée sur les trois aspects exposés ci-dessus. Principalement, on voulait démontrer l'efficacité des algorithmes de planification développés. Malheureusement, on s'est vite aperçu que, bien qu'ils donnent de bons résultats pour des petits problèmes (par exemple, le monde des blocs) les algorithmes de planification ont une performance plutôt médiocre lorsqu'on les applique à des problèmes du monde réel. En effet, comme la taille de l'espace de recherche des algorithmes de planification croît de façon exponentielle avec la complexité de la tâche pour laquelle on cherche à établir un plan, et que les tâches du monde réel sont des tâches très complexes, on obtient par conséquent, un espace de recherche extrêmement large, ce qui donne des temps de recherche proportionnellement longs.

Devant l'apparence d'échec des chercheurs en IA à développer des algorithmes de planification efficaces dans le monde réel, certains chercheurs ont commencé à questionner

les suppositions qui sont à la base de l'intelligence artificielle symbolique. Plus particulièrement, on émet de grandes réserves quant à l'approche *logicienne*¹ utilisée jusqu'à maintenant.

Agents réactifs : Un des grands noms parmi les critiques du raisonnement symbolique fut Brooks, qui, par le biais de plusieurs papiers [16,17,18], exposa son opposition au modèle symbolique et proposa une approche alternative qu'on appelle aujourd'hui IA réactive (SMA réactifs). Selon lui, le comportement intelligent devrait émerger de l'interaction entre divers comportements plus simples. Ainsi, au sein de son programme de recherche, il a développé l'architecture *subsumption*. Dans cette architecture, on bâtit des agents sans utiliser de représentation symbolique ni de raisonnement. Un agent est alors vu comme un ensemble de comportements accomplissant une tâche donnée. Chaque comportement est une machine à états finis qui établit une relation entre une entrée sensorielle et une action en sortie.

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

Bousquet et Lepage [19] reviennent en détails sur ce type d'approche généralement utilisée pour la simulation. Malgré la simplicité apparente et les bons résultats obtenus pour certaines applications [17,29,77,142], bien des reproches ont été adressés à cette approche dite "réactive", parmi lesquels, il convient de voir que :

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

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

interactions dynamiques entre les diverses couches deviennent trop complexes à comprendre.

Architectures hybrides : Dès le début des années 90, on savait que les systèmes réactifs pouvaient bien convenir pour certains types de problèmes et moins bien pour d'autres. De même, pour la plupart des problèmes, les solutions de l'IA classique, basée uniquement sur la planification, ne conviennent pas non plus. On commence dès lors à investiguer la possibilité de combiner les deux approches afin d'obtenir une architecture hybride [24,25,27,54,57,63]. Dans ce cas, un agent est composé de plusieurs couches, arrangées selon une hiérarchie, la plupart des architectures considérant que trois couches suffisent amplement. Ainsi, au plus bas niveau de l'architecture, on retrouve habituellement une couche purement réactive, qui prend ses décisions en se basant sur des données brutes en provenance des senseurs. La couche intermédiaire fait abstraction des données brutes et travaille plutôt avec une vision qui se situe au niveau des connaissances de l'environnement. Finalement, la couche supérieure se charge des aspects sociaux de l'environnement, c'est à dire du raisonnement tenant compte des autres agents.

Architectures BDI : Dans le cadre du raisonnement pratique, un raisonnement orienté vers la prise en compte des états mentaux, les chercheurs ont développé l'architecture BDI [14,15,63,110,111,112,126] (de l'anglais *Belief, Desire, Intention* pour croyance, désir, et intention), une architecture bâtie autour du *raisonnement pratique*². Ces agents sont généralement représentés par un "état mental" ayant les attitudes mentales suivantes :

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

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

1.1.2 Les applications des agents autonomes

Les programmes d'aujourd'hui interagissent avec leur usager selon un paradigme appelé manipulation directe. C'est-à-dire que le programme fait exactement ce que l'usager lui demande de faire. Cependant, il pourrait être intéressant d'avoir des programmes qui prennent l'initiative dans certaines circonstances. Ainsi, un logiciel prendrait maintenant une part active pour aider l'usager à accomplir ses tâches [95,96]. Des applications de tels agents se retrouvent principalement dans les lecteurs de courriels, les lecteurs de groupes de discussions actifs ainsi que les navigateurs actifs.

²Traduction du terme anglais *practical reasoning*, i.e., le raisonnement "pratique" tel qu'il se pratique chez les humains.

À part les agents interfaces, la technologie agent se trouve également au niveau (1) des agents mobiles [67,120] et (2) des agents émotionnels [9,10,70]. Le premier type d'agent est chargé de migrer d'une machine à une autre pour exécuter par exemple un code et ne pas surcharger le réseau. Le deuxième, quant à lui, a été conçu en vue de refléter les "émotions" comme le feraient des humains placés dans la même situation. Ce type d'agent s'est déjà avéré utile dans des applications du type animation par exemple [70].

En fait, l'agent en tant qu'entité individuelle peut s'avérer limité dans des bien des cas, surtout au vu de ce qu'on voit aujourd'hui comme applications distribuées et pour lesquelles un ensemble d'agents mettant en commun compétences et connaissances paraît plus que nécessaire. Ce type d'agents forment ce qu'on appelle des systèmes multiagents.

1.2 Les systèmes multiagents

Un *système multiagent* est un système distribué composé d'un ensemble d'agents. Contrairement aux systèmes d'IA, qui simulent dans une certaine mesure les capacités du raisonnement humain, les SMA sont conçus et implantés idéalement comme un ensemble d'agents interagissants, le plus souvent, selon des modes de *coopération*, de *concurrence* ou de *coexistence* [25,26,100].

Un SMA est généralement caractérisé par :

1. chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi chaque agent a un point de vue partiel;
2. il n'y a aucun contrôle global du système multiagent;
3. les données sont décentralisées;
4. le calcul est asynchrone.

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

Bien que les SMA offrent de nombreux avantages potentiels, ils doivent aussi relever beaucoup de défis. Voici les problèmes inhérents à la conception et à l'implémentation des SMA, d'après [11,58,72] :

1. Comment formuler, décrire, décomposer, et allouer les problèmes et synthétiser les résultats ?
2. Comment permettre aux agents de communiquer et d'interagir ? Quoi et quand communiquer ?
3. Comment assurer que les agents agissent de manière cohérente i) en prenant leurs décisions ou actions, ii) en gérant les effets non locaux de leurs décisions locales et iii) en évitant les interactions nuisibles ?
4. Comment permettre aux agents individuels de représenter et raisonner sur les actions, plans et connaissances des autres agents afin de se coordonner avec eux ? Comment raisonner sur l'état de leurs processus coordonnés (comme l'initialisation ou la terminaison) ?
5. Comment reconnaître et réconcilier les points de vue disparates et les intentions conflictuelles dans un ensemble d'agents essayant de coordonner leurs actions ?
6. Comment trouver le meilleur compromis entre le traitement local au niveau d'un seul agent et le traitement distribué entre plusieurs agents (traitement distribuée qui induit la communication) ? Plus généralement, comment gérer la répartition des ressources limitées ?
7. Comment éviter ou amoindrir un comportement nuisible du système global, comme les comportements chaotiques ou oscillatoires ?
8. Comment concevoir les plates-formes technologiques et les méthodologies de développement pour les SMA ?

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

- systèmes distribués [52,69];
- interface personnes-machines [81,103];
- bases de données et bases de connaissances distribuées coopératives [7];
- systèmes pour la compréhension du langage naturel [4,6,31,32,33,59,78,93];

- protocoles de communication et réseaux de télécommunications [12,69,98,108];
- programmation orientée agents et génie logiciel [76,123,134];
- robotique cognitive et coopération entre robots [86,87,88];
- applications distribuées comme le web, l'Internet, le contrôle de trafic routier, le contrôle aérien, les réseaux d'énergie, etc. [26,75].

Les recherches dans le domaine des systèmes multiagents poursuivent deux objectifs majeurs : Le premier concerne l'analyse théorique et expérimentale des mécanismes qui ont lieu lorsque plusieurs entités autonomes interagissent. Le seconde s'intéresse à la réalisation de programmes distribués capables d'accomplir des tâches complexes via la coopération et l'interaction. Leur position est donc double : d'un côté elles se placent au sein des sciences cognitives, des sciences sociales et des sciences naturelles pour à la fois modéliser, expliquer et simuler des phénomènes naturels, et susciter des modèles d'auto-organisation; de l'autre côté, elles se présentent comme une pratique, une technique qui vise la réalisation de systèmes informatiques complexes à partir des concepts d'agent, de communication, de coopération et de coordination d'actions.

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

En accord avec Jennings [76], nous pensons que la technologie agent va représenter durant les prochaines années, un nouveau paradigme de programmation pour le génie logiciel. Un nouveau paradigme "similaire" à la *programmation orientée objet* et qui d'ailleurs pourrait s'intituler *programmation orientée agent*.

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

Nous dressons maintenant un bref historique du domaine relatif aux SMA, en présentant des travaux précurseurs à ces systèmes ainsi que quelques exemples des premières applications développées.

1.2.1 Historique

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

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

Parmi les systèmes développés tout au début, il y avait :

Le modèle des acteurs : Un des premiers modèles proposés à l'époque de l'IAD fut le modèle des Acteurs [1,2]. Les acteurs sont des composantes autonomes d'un système qui communiquent par messages asynchrones. Ils sont composés d'un ensemble de primitives, parmi lesquelles on retrouve :

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

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

Le protocole Contract Net : Le protocole du *Contract Net* [42] fut une des premières solutions au problème d'allocation de tâches auquel fait face généralement un ensemble de solveurs de problèmes. Dans ce protocole, les agents peuvent prendre deux rôles: *gestionnaire* ou *contracteur*. L'agent qui doit exécuter une tâche donnée (le *gestionnaire*) commence tout

d'abord par décomposer cette tâche en plusieurs sous-tâches. Il doit ensuite annoncer les différentes sous-tâches au reste des agents de l'environnement. Les agents qui reçoivent une annonce de tâche à accomplir peuvent ensuite faire une proposition devant refléter leur capacité à remplir cette tâche. Le gestionnaire rassemble ensuite toutes les propositions qu'il a reçues et alloue la tâche à l'agent ayant fait la meilleure proposition.

Les premières applications : Parmi les premières applications développées à l'aide de la technologie multiagents, on retrouve une application dans le contrôle du trafic aérien et une autre dans la surveillance de véhicules motorisés. Pour le contrôle de trafic aérien, Cammarata [23] a étudié des stratégies de coopération pour pouvoir résoudre les conflits entre les plans d'un groupe d'agents. Ces stratégies ont pu être ensuite appliquées dans le cadre d'un système de contrôle du trafic aérien. Dans ce système, chaque agent (représentant un avion) cherche à bâtir un plan de vol qui devrait le garder à une distance sécuritaire de chaque autre agent. Dans le cas où des agents se retrouveraient dans une situation conflictuelle, ils doivent alors choisir parmi eux, un agent apte à élaborer un nouveau plan de vol sans engendrer de nouveaux conflits. Pour faire ce choix, Cammarata et ses collègues ont opté pour deux stratégies qu'ils ont comparées entre elles. La première a consisté à choisir l'agent le mieux informé ou l'agent le moins contraint pour jouer le rôle de planificateur central et élaborer un nouveau plan de vol qui résoudrait le conflit. La deuxième a consisté à faire le partage suivant: il revient à l'agent le mieux informé d'élaborer un nouveau plan de vol et à l'agent le moins contraint de l'exécuter.

La surveillance de véhicules motorisés tournait, quant à elle, autour du système DVMT (Distributed Vehicle Monitoring Task) [39]. Ce système avait comme tâche principale d'identifier quel type de véhicule circulait dans la zone où étaient placés ses senseurs. À partir de ces interprétations, il devait tenter d'établir une description des mouvements des véhicules dans la région qu'il supervisait. Comme bien d'autres systèmes multiagents de l'époque, le système DVMT utilisait un *blackboard* pour coordonner les efforts de ses différents agents. Rappelons qu'un *blackboard* [48] est simplement une structure de données partagées entre divers agents. Ces derniers peuvent la consulter pour obtenir des informations sur l'état actuel du problème ou y écrire la partie de la solution qu'ils ont obtenue.

2. Interactions et coopération entre agents

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

Dans le cadre des systèmes multiagents on doit traiter la question fondamentale qui est l'implantation de la coopération entre les agents [44]. Durfee et ses collègues [47] ont proposé

quatre buts génériques pour établir la coopération dans un groupe d'agents: augmenter le taux de finalisation des tâches grâce au parallélisme; augmenter le nombre de tâches réalisables grâce au partage de ressources (information, expertise, dispositifs physiques, etc.); augmenter les chances de finaliser des tâches en les dupliquant et en utilisant éventuellement des modes de réalisation différents; diminuer les interférences entre tâches en évitant les interactions négatives.

En fait, on peut caractériser un système par le type de coopération mis en oeuvre qui peut aller de la coopération totale à l'antagonisme total [100]. Des agents totalement coopératifs peuvent changer leurs buts pour répondre aux besoins des autres agents afin d'assurer une meilleure coordination entre eux. Cela peut résulter en des coûts de communication élevés.

Les agents antagonistes par contre, ne vont pas coopérer et dans ce cas, leurs buts respectifs vont se trouver bloqués. Dans de tels systèmes, les coûts de communication sont minimaux. La plupart des systèmes réels se situe entre les deux extrêmes : coopération totale et antagonisme total. La coopération totale est mise en oeuvre par exemple dans les approches de résolution coopérative de problèmes distribués ("cooperative distributed problem solving" ou CDPS) proposée par Durfee et ses collègues [47]. Dans ces approches, les agents coopèrent pour résoudre des problèmes qu'ils ne peuvent pas résoudre individuellement. Tout d'abord chaque agent utilise ses connaissances et ressources pour résoudre localement un ou plusieurs sous-problèmes. Les solutions partielles à tous les sous-problèmes sont par la suite intégrées. C'est par exemple le cas des avions qui volent dans un même espace aérien et qui, de ce fait, ont différentes perceptions de leur espace environnant obtenues par leurs senseurs respectifs. Ce n'est qu'en combinant ces différentes perceptions que ces avions peuvent obtenir une vue globale afin de résoudre les conflits potentiels qui peuvent se produire entre eux.

Ferber [53] remarque que les chercheurs ont développé différents points de vue sur la coopération. Ainsi, on peut considérer la coopération comme une attitude adoptée par les agents qui décident de travailler ensemble ou on peut adopter le point de vue d'un observateur extérieur au système multiagent qui interprète a posteriori les comportements des agents pour les qualifier de coopératifs ou non suivant des critères préétablis tels que l'interdépendance des actions ou le nombre de communications effectuées. Dans le cas de la coopération vue comme une attitude intentionnelle, les agents s'engagent dans une action après avoir identifié et adopté un but commun. C'est par exemple le cas d'un groupe de personnes acceptant de travailler dans une entreprise. Des chercheurs comme Galliers [59] et Conte [38] ont souligné l'importance de l'adoption d'un but commun pour ce type de coopération qu'ils considèrent comme un élément essentiel de l'activité sociale.

2.1 Interactions entre agents coopératifs

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

La planification pour un seul agent, telle que discutée à la section 1.1.1, ne fait que construire une séquence d'actions en ne considérant que les buts de l'agent, ses capacités et les contraintes imposées par son environnement. Par contre, dans un environnement multiagent, on se doit de

tenir compte des contraintes que les actions des autres agents placent sur le choix des actions de l'agent; des contraintes que les engagements de l'agent envers les autres imposent sur son propre choix d'action; ainsi que l'évolution imprévisible de l'environnement due à des agents qui ne sont pas modélisés.

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

Une autre approche intéressante pour résoudre les interdépendances entre les sous-buts a été le "*Functionally Accurate Model*" [89,90] dans lequel les agents n'ont pas besoin d'avoir toutes les informations nécessaires pour résoudre les conflits localement. Les interactions se font plutôt par un échange asynchrone de résultats partiels. À partir de ce modèle sont apparus plusieurs principes de coordination, parmi lesquels on retrouve le *Partial Global Planning* [45,46].

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

Un autre courant de recherche sur la coordination des agents consiste à modéliser explicitement le travail d'équipe qu'effectuent les agents. Cette approche s'avère très pratique dans des environnements dynamiques où les membres de l'équipe peuvent échouer dans leur tâches ou découvrir de nouvelles opportunités. Dans de telles situations, l'équipe doit pouvoir évaluer sa performance et être en mesure de se réorganiser en conséquence. Dans cet ordre d'idées, on parle alors du modèle des "intentions conjointes" [73,92]. Un tel modèle est d'ailleurs une extension naturelle du raisonnement pratique (comme vu en section 1.1.1). Dans ce modèle, on cherche à modéliser l'état mental d'une équipe en action. L'équipe aura une intention conjointe si chaque membre de l'équipe est engagé à compléter l'action de l'équipe. Préalablement à cette intention collective, tous les membres de l'équipe auront établi les croyances et les engagements communs appropriés, et ce en s'échangeant des actes (du langage naturel) de requête et de confirmation.

2.2 Interactions entre agents égo-centrés

Les interactions entre agents égo-centrés se basent principalement sur la négociation. Cette dernière devient donc une méthode de coordination et de résolution de conflits. Elle a également été utilisée comme métaphore pour la communication de changements dans les plans, l'allocation de tâches ainsi que la résolution centralisée de violation de contraintes. On peut donc voir que la définition de la négociation est aussi imprécise que celle du concept d'agent. On arrive malgré tout à retrouver certaines caractéristiques importantes de la négociation:

- la présence d'une forme de conflit qui doit être résolu de façon décentralisée; par
- des agents égo-centrés; ayant
- une rationalité limitée; et
- des informations incomplètes.

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

Le système PERSUADER de Sycara [130,131,132,133] et les travaux de Rosenschein [115,117] sont les premiers travaux de l'intelligence artificielle distribuée sur la négociation entre agents égo-centrés. Le système PERSUADER puise son modèle dans les négociations de conventions collectives. Les négociations impliquent donc trois agents (un syndicat, une compagnie et un médiateur) et s'inspirent des négociations entre êtres humains. On y modélise les multiples itérations pendant lesquelles les parties impliquées s'échangent offres et contre-offres, et l'utilité des agents est multidimensionnelle et privée. Les croyances des agents sont révisées de telle sorte qu'on peut atteindre une entente grâce à une argumentation persuasive.

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

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

Krauss [83,84] de son côté s'est penché sur le rôle du temps dans les négociations. Cette étude lui a permis de voir que l'importance qu'un agent accorde au temps peut affecter l'entente finale entre les parties.

Comme le commerce électronique devient de plus en plus une réalité, le besoin de techniques de négociations qui tiennent compte des complexités du monde réel grandit. On se doit donc de tenir compte des informations incomplètes, des problèmes de négociations multiples, des échéances pour les négociations et la possibilité de briser des contrats. On a donc ajouté la possibilité d'établir des contrats moins rigides au sein du protocole *Contract Net*, des pénalités étant attribuées lorsqu'un contrat n'est pas rempli complètement [121].

3. Coordination entre agents

De nombreux exemples de coordination existent dans la vie quotidienne : deux déménageurs déplaçant un meuble lourd, deux jongleurs échangeant des balles avec lesquelles ils jonglent, des personnes qui parlent à tour de rôle en se passant un micro, etc. Malone [97] note que deux des composantes fondamentales de la coordination entre agents sont l'allocation de ressources rares et la communication de résultats intermédiaires. Dans ce contexte, les agents doivent être capables de communiquer entre eux de façon à pouvoir échanger les résultats intermédiaires. Pour l'allocation des ressources partagées, les agents doivent être capables de faire des transferts de ressources. Ceci peut d'ailleurs imposer certains comportements à des agents particuliers. Malone conclut qu'il peut être utile de distinguer les liens de contrôle comme une catégorie spéciale de liens de communication par lesquels certains agents transmettent des instructions que d'autres vont être motivés à suivre.

En étudiant les communautés humaines, Mintzberg [99] a identifié trois processus fondamentaux de coordination : ajustement mutuel, supervision directe et coordination par standardisation. L'*ajustement mutuel* est la forme de coordination la plus simple qui se produit quand deux ou plusieurs agents s'accordent pour partager des ressources en vue d'atteindre un but commun. Habituellement les agents doivent échanger de nombreuses informations et faire plusieurs ajustements à leurs propres comportements en tenant compte des comportements des autres agents. Dans cette forme de coordination aucun agent n'a un contrôle sur les autres agents et le processus de décision est conjoint. La coordination dans les groupes de pairs ("peer groups") et dans les marchés ("market") est habituellement une forme d'ajustement mutuel. La *supervision directe* apparaît quand un ou plusieurs agents ont déjà établi une relation dans laquelle un des agents a un contrôle sur les autres. Cette relation est habituellement établie par ajustement mutuel comme par exemple dans le cas d'un employé ou d'un sous-contractant qui accepte de suivre les instructions du superviseur. Dans cette forme de coordination l'agent superviseur contrôle l'utilisation des ressources partagées (comme par exemple les ressources humaines, le temps de calcul ou l'argent) par les agents subordonnés. Il peut aussi imposer certains comportements. Dans les cas de *coordination par standardisation* le superviseur coordonne les activités en établissant des procédures que doivent suivre les subordonnés dans des situations identifiées. On trouve par exemple de telles procédures dans les entreprises, mais aussi dans les systèmes informatiques.

Malone [97] suggère qu'en utilisant ces processus de coordination fondamentaux "à la Mintzberg", il est possible de construire des systèmes de coordination sophistiqués dont les plus répandus sont les hiérarchies et les marchés basés respectivement sur les processus de supervision directe et d'ajustement mutuel. Le travail en petit groupe se prête bien à l'ajustement mutuel. Par contre, dès que la taille du groupe grandit et que le nombre de tâches augmente, le nombre de liens et la quantité d'informations échangées peuvent devenir rapidement un handicap sérieux. Un groupe important peut être partagé efficacement en sous-groupes de manière à ce que la plupart des interactions s'effectuent dans les sous-groupes et que les quelques interactions nécessaires entre les sous-groupes soient prises en charge par les superviseurs de ces sous-groupes. Les sous-groupes peuvent être coordonnés par contrôle hiérarchique ou par ajustement mutuel suivant les caractéristiques du domaine d'application et des tâches à accomplir. Les marchés sont aussi considérés comme une activité d'organisation de groupe basée sur l'ajustement mutuel dans laquelle chaque agent contrôle des ressources rares (par exemple la

main d'oeuvre, les matières premières, les marchandises et l'argent). Les agents s'accordent pour partager leurs ressources respectives afin d'atteindre un but commun. Les ressources échangées ont un prix explicite ou implicite. Lorsqu'un contrat est conclu, il y a un accord pour que l'agent contracteur devienne le superviseur de l'agent contractant. De nombreuses organisations mettent en oeuvre des processus de coordination mixtes basées sur l'ajustement mutuel, la supervision directe et la standardisation. Ces cadres organisationnels peuvent être des sources d'inspiration quand on doit établir une structure organisationnelle pour un SMA.

Bien entendu, la coordination est une question centrale pour les SMAs et la résolution de systèmes distribués. En effet, sans coordination un groupe d'agents peut dégénérer rapidement en une collection chaotique d'individus. On pourrait penser que la façon la plus simple de s'assurer un comportement cohérent du groupe d'agents serait de le faire par un agent centralisateur qui détiendrait des informations de haut niveau sur ces agents. Ainsi, l'agent centralisateur pourrait créer des plans d'action et assigner les tâches aux divers agents du groupe. Cette approche est pratiquement impossible à mettre en oeuvre dans des applications réalistes en raison de la difficulté de réaliser un tel agent centralisateur qui puisse tenir compte des buts, des connaissances et des activités de chaque agent : la charge en communication serait énorme, sans compter qu'on perdrait les avantages d'un SMA composé d'agents autonomes [38]. Le contrôle et les informations doivent alors être distribués parmi les agents.

À cette fin, Jennings [73] propose un modèle intéressant qui se base sur l'hypothèse de la centralité des engagements et des conventions. Les engagements sont vus comme des promesses en vue de réaliser certaines actions, alors que les conventions constituent le moyen de faire le suivi de ces engagements dans des circonstances changeantes. Grâce aux engagements des autres agents, un agent peut prédire les actions qu'il est susceptible d'effectuer et déterminer ainsi les interdépendances avec ses propres actions. Mais comme le monde extérieur et les croyances des agents évoluent constamment, un agent doit aussi posséder un moyen pour déterminer si les engagements existants sont encore valides. Les conventions offrent un tel mécanisme en définissant les conditions sous lesquelles les engagements doivent être réévalués et en spécifiant les actions à entreprendre dans de telles circonstances.

Du point de vue du concepteur d'un SMA diverses questions doivent être traitées comme notamment [53] : Avec quels agents un agent doit-il coordonner ses actions? Quand et où ces actions de coordination doivent-elles être accomplies? Comment détecter et traiter les interactions entre actions (conflits et renforcement)? Détecter les relations existant entre des actions est une activité nécessaire lorsqu'on veut coordonner ces actions.

Von Martial [140] de son côté, a identifié deux grandes catégories de relations pouvant exister entre les actions accomplies simultanément par plusieurs agents : les relations négatives et les relations positives. Les relations négatives (ou conflictuelles) sont celles qui gênent ou empêchent plusieurs actions de s'accomplir simultanément et sont dues en général à des incompatibilités de buts ou des conflits de ressources. Par exemple dans une vente aux enchères, agent X et agent Y veulent acquérir un même meuble M. Les relations positives (ou synergiques) sont celles qui permettent aux actions de bénéficier les unes des autres. Ainsi, la réalisation d'une action a accomplie par l'agent Y réalise du même coup une action $x = b$ que devait accomplir l'agent X ou favorise la réalisation d'une action c par l'agent Z. Par exemple X, Y et Z sont dans une

pièce dont les fenêtres sont fermées et les stores baissés. X a chaud et Z aimerait avoir de la lumière. Y monte les stores et ouvre une fenêtre.

Le lecteur, intéressé par une étude détaillée de la coordination entre agents, pourra se référer à l'article de El Fallah-Seghrouni au chapitre ?? de ce volume [49]. Dans ce chapitre, la coordination est vue plus particulièrement sous l'angle de la coordination d'actions par planification multiagent.

4. Négociation entre agents

La négociation joue un rôle fondamental dans les activités de coopération en permettant aux personnes de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. Durfee et ses collègues [47] définissent la négociation comme le processus d'améliorer les accords (en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes. En général les chercheurs en IA distribuée utilisent la négociation comme un mécanisme pour coordonner un groupe d'agents. Différentes approches ont été développées en s'appuyant sur la riche diversité des négociations humaines dans divers contextes [36,42,117,131].

Un des protocoles les plus étudiés pour la négociation s'appuie sur une métaphore organisationnelle [42]. Le protocole du réseau contractuel ("Contract-Net") a été une des approches les plus utilisées pour les SMA [129]. Les agents coordonnent leurs activités grâce à l'établissement de contrats pour atteindre des buts spécifiques. Un agent, agissant comme un gestionnaire ("manager") décompose son contrat (une tâche ou un problème) en sous-contrats qui pourront être traités par des agents contractants potentiels. Le gestionnaire annonce chaque sous-contrat sur un réseau d'agents. Les agents reçoivent et évaluent l'annonce. Les agents qui ont les ressources appropriées, l'expertise ou l'information requise envoient au gestionnaire des soumissions ("bids") qui indiquent leurs capacités à réaliser la tâche annoncée. Le gestionnaire évalue les soumissions et accorde les tâches aux agents les mieux appropriés. Ces agents sont appelés des contractants ("contractors"). Enfin, gestionnaires et contractants échangent les informations nécessaires durant l'accomplissement des tâches. Par exemple, Parunak [109] a utilisé ce protocole pour développer un système de contrôle de production.

Un autre important protocole de négociation a été proposé par Cammarata et ses collègues [22,23] qui ont étudié les stratégies de coopération pour résoudre des conflits entre des plans d'un ensemble d'agents. Ces stratégies ont été appliquées au domaine du contrôle de trafic aérien avec le but de permettre à chaque agent (représentant un avion) de construire un plan de vol qui permettrait de garder une distance sécuritaire par rapport aux autres avions et de satisfaire des contraintes telles que "atteindre la destination désirée avec une consommation de carburant minimale". La stratégie choisie, appelée "centralisation de tâche" permettait aux agents impliqués dans une situation conflictuelle potentielle (des avions se rapprochant trop compte tenu de leurs caps respectifs) de choisir l'un d'eux pour résoudre le conflit. Cet agent agissait comme un planificateur centralisé et développait un plan multiagent qui spécifiait les actions concurrentes de tous les avions impliqués. Les agents utilisaient la négociation pour déterminer qui était le plus apte à réaliser le plan. Cette aptitude était évaluée à partir de divers critères permettant d'identifier par exemple l'agent le mieux informé ou celui qui était le plus contraint. Les

protocoles de négociation précédents supposent que les agents sont coopératifs, et donc qu'ils poursuivent un but commun.

Sycara [131] a étudié des situations dans lesquelles on ne peut pas supposer la coopération des agents. La résolution de conflits est mise en oeuvre soit par négociation, soit à l'aide d'un médiateur. Sycara a développé un système qui résout des conflits dans le domaine des relations de travail. Ce système comprenait trois agents : l'employeur, les employés et le médiateur. L'employeur et les employés avaient des buts conflictuels et le médiateur générait des propositions et contre-propositions pour essayer de rapprocher les deux parties adverses.

Müller de son côté [102], a indiqué que les recherches en négociation peuvent être divisées en trois catégories. Les recherches sur les langages de négociation qui s'intéressent aux primitives de communication pour la négociation, à leur sémantique et à leur usage dans les protocoles. Les recherches sur les décisions en négociation qui s'intéressent, avant à eux, aux algorithmes pour comparer les sujets de négociation, les fonctions d'utilité et la caractérisation des préférences des agents. Les recherches sur le processus de négociation qui étudient des modèles généraux des comportements de négociation des agents.

5. Planification dans un environnement multiagent

Bond et Gasser [11] font remarquer que nous pouvons obtenir une plus grande coordination si on oriente les comportements des agents vers des buts communs en établissant explicitement une division du travail entre les agents. Des techniques comme la planification centralisée pour des groupes d'agents, la conciliation de plans, la planification distribuée, l'analyse organisationnelle, sont toutes des façons d'aider les agents à aligner leurs activités en assignant les tâches après avoir raisonné sur les conséquences de réaliser ces tâches dans des ordres particuliers. Dans une approche de planification multiagent, un plan multiagent est un plan qui est créé afin que plusieurs agents puissent l'exécuter. La création d'un tel plan peut être faite par un seul ou par plusieurs agents. En planification multiagent centralisée un agent est responsable pour la création du plan qui spécifie les actions planifiées pour tous les agents concernés. C'est cette approche qui a été utilisée par Cammarata pour le problème de contrôle de trafic aérien.

Une autre façon d'implémenter la planification multiagent centralisée a été proposée par Georgeff [62]. Dans cette approche, les plans des agents sont d'abord créés de façon individuelle; ensuite un agent centralisateur rassemble ces plans et les analyse pour identifier les conflits. L'agent centralisateur en question essaye de résoudre les conflits en modifiant les plans locaux des autres agents et en introduisant des commandes de communication afin que les agents se synchronisent de façon appropriée. Dans une approche de planification distribuée, les activités de planification sont réparties au sein d'un groupe d'agents. Cette approche est utilisée quand un seul agent ne peut pas avoir une vue globale des activités du groupe.

Von Martial [140] a distingué, quant à lui, deux classes de problèmes en planification distribuée : la planification dirigée par les tâches et la coordination de plans. Dans les systèmes de planification dirigés par les tâches, il y a un but initial ou une tâche initiale qui est décomposée en sous-buts ou sous-tâches et qui est répartie entre plusieurs agents (décomposition de problème "top-down"). Par ailleurs, la coordination de plan s'attaque à des situations dans lesquelles il existe des plans d'agents (vision centrée sur les agents) et le problème consiste à réconcilier ces

plans avant de les exécuter dans un environnement commun. En général, la planification multiagent nécessite une forme ou une autre de synchronisation de plans qui peut être réalisée à divers moments : pendant la décomposition de plan, pendant la construction de plan ou après celle-ci. Les plans des agents peuvent être en conflits en raison d'incompatibilités d'états des systèmes, de l'ordre des activités ou de l'usage des ressources. De tels conflits peuvent être résolus par un agent en particulier (coordonnateur ou médiateur) ou une solution peut être obtenue par négociation.

Par la suite, Von Martial [137] a proposé un modèle de coordination de plans qui s'appuie sur la résolution de conflits basée sur les divers types de relations (négatives et positives) pouvant exister entre ces plans d'une part, et sur un protocole de communication pour la synchronisation des plans d'autre part. Les relations entre plans entraînent une modification des plans des agents. Les relations négatives sont traitées pour résoudre les conflits et les relations positives sont utilisées pour tirer parti de plans convergents.

Conry et ses collègues [36,37] ont développé un protocole de négociation appelé "multi-stage negotiation" pour résoudre de façon coopérative des conflits dans l'allocation de ressources. Le domaine d'application était celui de la surveillance et du contrôle dans un système de communication complexe. Le protocole débute par la génération d'un plan initial et consiste en plusieurs cycles permettant d'envoyer des requêtes pour des buts secondaires, l'analyse locale, la génération de plans alternatifs et l'envoi de réponses.

Durfee and Lesser [45,47] ont proposé une approche appelée "planification partielle globale" (PGP pour Partial Global Planning) dans laquelle les agents construisent des plans et partagent ces plans pour identifier des améliorations potentielles pour leur coordination. Au contraire de la planification multiagent qui suppose qu'un plan est élaboré avant que les agents commencent à agir, la planification partielle globale permet aux agents d'alterner phases de planification et d'exécution : les agents commencent par planifier des interactions coordonnées au mieux de leurs connaissances et réagissent aux situations non prévues en modifiant leurs plans. Cette approche a été appliquée à une plate-forme de tests multiagents appelée DVMT ("Distributed Vehicle Monitoring Testbed"). Dans ce contexte, Lesser [90] a utilisé le modèle FA/C "Functionally Accurate / Cooperative mode") qu'il a en fait, développé pour répondre aux déficiences des modèles conventionnels qui supposent que les tâches sont décomposées de façon à ce que chaque agent ait suffisamment de données pour résoudre complètement et précisément les sous-problèmes qui lui sont assignés, et ce avec peu d'interaction avec les autres agents. Toutefois, dans de nombreuses applications (interprétation distribuée, planification distribuée et allocation de ressources, etc.) ce modèle de décomposition de tâches est inadéquat. Dans le modèle FA/C les agents n'ont pas besoin de disposer de toutes les informations localement pour résoudre leurs sous-problèmes. Ils interagissent en échangeant des résultats partiels. Dans un SMA les agents doivent gérer des ressources distribuées qui peuvent être physiques (capacités de communication, matières premières, argent) ou informationnelles (telles que les informations au sujet de la décomposition du problème). Les agents doivent adapter leurs plans pour tenir compte de la disponibilité des ressources.

Biern entendu, la planification contribue à la coordination, dans la mesure où lorsque les agents adoptent un plan « bien fait », ils agissent généralement de manière coordonnée. El-Fallah [50,51] a travaillé sur cet aspect dont les éléments essentiels sont repris au chapitre ??.

6. Communication entre agents

Les agents peuvent interagir soit en accomplissant des actions linguistiques (en communiquant entre eux), soit en accomplissant des actions non-linguistiques qui modifient leur environnement. En communiquant, les agents peuvent échanger des informations et coordonner leurs activités. Dans les SMA deux stratégies principales ont été utilisées pour supporter la communication entre agents: les agents peuvent échanger des messages directement ou ils peuvent accéder à une base de données partagées (appelée tableau noir ou “blackboard”) dans laquelle les informations sont postées. Les communications sont à la base des interactions et de l’organisation sociale d’un SMA.

Dans certains cas, un agent peut inférer les plans des autres agents sans avoir à communiquer avec eux. Dans Tubbs [136] ce mécanisme est appelé “marchandage tacite” et il semble fonctionner au mieux quand les buts des agents ne sont pas conflictuels. Pour étudier ce mécanisme, Rosenschein et ses collègues [113] ont utilisé une approche de la théorie des jeux caractérisée par des matrices de gains qui contiennent les gains des agents pour chacune des issues possibles de l’interaction. Cette approche suppose que les agents partagent la connaissance de cette matrice de gains, ce qui n’est pas réaliste si les agents ne sont pas bienveillants. Rosenschein [115] a étendu cette approche afin de permettre aux agents de raffiner leurs choix en tenant compte de l’incertitude des choix des autres agents. Cette approche présente aussi des difficultés car le fait que les agents aient à spéculer sur les choix des autres agents conduit à développer des mécanismes de raisonnement complexes qui peuvent nécessiter une quantité de calculs importante. Il est clair que l’absence de communication doit être compensée par des mécanismes de raisonnement plus complexes. Certains chercheurs ont examiné l’utilisation de communications primitives restreintes à l’utilisation d’un ensemble fini de signaux (souvent au nombre de deux) avec des interprétations fixes “à la Hoare”. Cette stratégie a été employée par Georgeff [62] pour éviter des conflits dans une planification multiagent. Cependant, des informations plus sophistiquées comme des requêtes ou des commandes ne peuvent pas être exprimées avec de tels signaux.

6.1 Transfert de plans ou de messages

Dans l’approche de transfert de plans, un agent X communique son plan en totalité à un agent Y et Y communique son plan en totalité à X . Cette approche présente plusieurs inconvénients : le transfert de plans est coûteux en ressources de communication et c’est une approche difficile à mettre en oeuvre dans des applications réelles car il y a en général un fort degré d’incertitude au sujet des états actuel et futurs du monde. Dans des situations réelles en plus, il n’est pas possible de formuler à l’avance des plans en totalité. Par conséquent, on a besoin de communiquer des stratégies générales aux agents plutôt que des plans, et donc il est nécessaire que les agents puissent échanger des messages. De très nombreux travaux ont mis en oeuvre le transfert de message associé à l’usage de protocoles précis. Dans un SMA la complexité des communications dépend des caractéristiques des agents. Des agents réactifs utilisent un ensemble de règles de communication qui sont déclenchées quand des états spécifiques sont vérifiés ou quand des événements prédéterminés se produisent. Ces événements peuvent être des messages reçus ou des changements dans l’environnement. Les types de messages combinés avec les réponses attendues conduisent à la spécification de protocoles d’interaction. Comme on l’a vu, de tels protocoles ont été développés pour divers types de systèmes comme le protocole du réseau contractuel [128] ou

des protocoles de coopération [23]. Les communications et l'exécution d'actions sont en général intercalées. Aussi des agents intentionnels doivent être capables de construire des plans dans lesquels des actions linguistiques et non linguistiques sont combinées. Kreifelts et Von Martial [85] ont proposé un protocole de négociation de ce type.

6.2 Échange d'informations grâce à un tableau noir

En intelligence artificielle la technique du tableau noir ("blackboard") est très utilisée pour spécifier une mémoire partagée par divers systèmes [105,106]. Dans un SMA utilisant un tableau noir, les agents peuvent écrire des messages, insérer des résultats partiels de leurs calculs et obtenir de l'information. Le tableau noir est en général partitionné en plusieurs niveaux qui sont spécifiques à l'application. Les agents qui travaillent sur un niveau particulier peuvent accéder aux informations contenues dans le niveau correspondant du tableau noir ainsi que dans des niveaux adjacents. Ainsi, les données peuvent être synthétisées à n'importe quel niveau et transférées aux niveaux supérieurs alors que les buts de haut niveau peuvent être filtrés et passés aux niveaux inférieurs pour diriger les agents qui oeuvrent à ces niveaux. Le transfert de messages et les communications basées sur un tableau noir sont souvent combinés dans des systèmes complexes. Dans de tels systèmes, chaque agent est composé de plusieurs sous-systèmes (ou "sous agents") qui communiquent à travers un tableau noir. Les agents communiquent entre eux par échange de messages. Plusieurs MAS ont été bâtis sur ce genre d'architecture comme le DVMT (Distributed Vehicle Monitoring Testbed) [89] ou le système MINDS [71]. Dans le système MINDS chaque usager travaille avec un agent qui utilise un tableau noir pour retrouver des documents dans sa base de données locales ou communique par messages avec d'autres agents du même type.

6.3 Actes de discours et conversations

Dès lors que l'on s'intéresse au contenu des messages échangés par les agents, il est pertinent d'examiner les nombreux travaux qui ont été réalisés sur la compréhension du langage naturel et en particulier les recherches sur l'intentionnalité dans les communications [32]. Dans ces travaux on considère que les agents sont dotés de structures de données appelées "états mentaux" à partir desquelles ils peuvent raisonner. Diverses catégories d'états mentaux ont été retenues par les chercheurs dont notamment les croyances, les désirs, les intentions ou buts, les capacités, etc. Ce sont ces mêmes états mentaux qui caractérisent les modèles des agents cognitifs en intelligence artificielle distribuée. Ainsi, un agent cognitif raisonne sur ses croyances qui représentent sa compréhension du monde dans lequel il évolue. Il raisonne aussi sur ses désirs et intentions en relation avec ses croyances et capacités afin de prendre des décisions auxquelles sont associés les plans qu'il va accomplir pour agir dans le monde. Lorsqu'un agent *X* communique avec un autre agent *Y*, c'est pour influencer les états mentaux de *Y*. Ainsi, *X* peut transférer à *Y* une information qui va changer les croyances de *Y* ou bien lui proposer d'adopter un but que *Y* va transformer en une des ses intentions. Il est donc pertinent de s'intéresser à la façon dont les messages échangés peuvent être structurés pour refléter ces mécanismes d'influence sur les états mentaux des agents.

Les philosophes du langage ont développé au cours des 25 dernières années la théorie des actes de discours qui considère que "dire quelque chose c'est en quelque sorte agir" [122]. Les paroles sont interprétées comme des actions appelées actes de discours. Cette théorie fournit un cadre qui identifie des types d'actes de discours primitifs et permet de dériver de ces types

primitifs l'interprétation sous forme logique de n'importe quel verbe du langage qui exprime un acte de discours quelconque. Les chercheurs en intelligence artificielle ont très tôt pensé à étudier comment les actes de discours peuvent être utilisés pour influencer les états mentaux des agents. Cohen et Perrault ont travaillé sur la reconnaissance et la génération d'actes de discours en utilisant des techniques de planification [31].

Allen [4] suggéra qu'un locuteur peut identifier des actes de discours directs et indirects s'il est capable de reconnaître le plan de son interlocuteur ainsi que les obstacles qui empêchent sa réalisation. Cependant, cette approche nécessite que l'agent soit capable d'identifier avec précision les catégories d'actes de discours correspondant aux énonciations de ses interlocuteurs, ce qui n'est pas toujours facile comme le remarquent Cohen et Levesque [33]. Cohen et Levesque [31,32] ont proposé une théorie permettant de tenir compte des états mentaux dans le processus de communication. Ils partent d'une théorie des interactions rationnelles qui se base sur les intentions et engagements des agents. Cette approche qui a influencé grandement les recherches dans ce domaine suppose que les agents sont bienveillants. Galliers [59] a étendu cette théorie à des situations mettant en jeu des agents qui peuvent être en conflit, en montrant qu'il n'est pas nécessaire de considérer que les agents sont bienveillants. Parallèlement Werner [143] a proposé une théorie unifiant des techniques de communication, de coopération et de structuration sociale pour la conception de systèmes d'agents qui adoptent des comportements de groupe : une formalisation des états intentionnels des agents en relation avec les actes de discours et leurs effets sur le processus de planification.

Shoham [124] a proposé une approche formelle décrivant diverses lois sociales applicables aux sociétés d'agents. Castelfranchi et Conte [21] soulèvent diverses objections aux fondements sociologiques actuels de l'IAD : une emphase trop grande est mise sur la coopération tant au niveau des buts partagés, des actions conjointes et du partage d'information; les agents sont "hypercognitifs" dans le sens que l'on dote les agents de capacités d'omniscience, d'introspection qui ne sont pas socialement plausibles; on néglige les mécanismes équivalents à des lois, règlements ou normes qui limitent la liberté des agents; on néglige aussi le rôle de la communication inter-agents comme moyen d'influencer les autres. Ces auteurs proposent une voie alternative dans laquelle on mettrait l'emphase sur les dépendances sociales existant entre agents et sur leur influence dans la formation de buts communs. Pour adopter un comportement social, un agent doit être capable non seulement de raisonner sur ses états mentaux mais aussi de prendre en compte les connaissances qu'il partage avec les autres agents ("mutual belief") et les engagements qui les relient [33,111,126]. On mesure ainsi l'importance d'associer à un modèle de la communication inter-agents un modèle de raisonnement sur les états mentaux, tout en tenant compte de la dimension sociale de l'interaction. Les relations sociales ont aussi un impact évident dans les conversations humaines : la forme et le contenu des actes de discours des locuteurs tiennent compte des rôles et statuts sociaux des participants à la conversation.

Cet aspect n'a été exploré que récemment dans le cadre de la modélisation des interactions et la simulation des conversations entre agents logiciels [13]. Barbuceanu et Fox [8] considèrent une activité de coordination entre agents comme un genre de "conversation" qui est décrite par un graphe d'états finis : les états du graphe représentent les états que peut prendre la conversation. Un ensemble de règles de conversation spécifient comment un agent qui est dans un état conversationnel donné reçoit un message d'un type particulier (spécifié suivant les conventions

de KQML), réalise des actions locales (ex. “mettre à jour sa base de données locale”), envoie des messages et passe dans un autre état conversationnel.

De leur côté, Rousseau et Moulin [119] ont proposé une approche de simulation des conversations entre agents vue comme un processus de négociation multi-niveaux au cours duquel les participants négocient au sujet de leurs états mentaux (croyances, désirs, intentions, émotions, etc.). Dans cette approche les actes de discours sont interprétés par les agents en termes d’objets de conversation (OCs) qui sont des états mentaux associés à des positionnements d’agents (“proposition”, “acceptation”, “refus”). Les OCs proposés par les agents locuteurs au cours de la conversation forment un réseau de concepts qui constitue une mémoire persistante de la conversation partagée par les agents locuteurs et à partir de laquelle ils peuvent raisonner.

6.4 Communication utilisant KQML, ACL et les conversations

Le langage KQML [56] a été proposé pour supporter la communication inter-agents. Ce langage définit un ensemble de types de messages (appelés abusivement “performatifs”) et des règles qui définissent les comportements suggérés pour les agents qui reçoivent ces messages. Les types de messages de KQML sont de natures diverses : simples requêtes et assertions (ex. “ask”, “tell”); instructions de routage de l’information (“forward” et “broadcast”); commandes persistantes (“subscribe”, “monitor”); commandes qui permettent aux agents consommateurs de demander à des agents intermédiaires de trouver les agents fournisseurs pertinents (“advertise”, “recommend”, “recruit” and “broker”). Comme le font remarquer Cohen et Levesque [34], ce langage a été développé de façon ad-hoc pour les besoins des développeurs d’agents logiciels : le terme “performatif” a été utilisé pour nommer diverses commandes qui ont une certaine ressemblance avec des verbes utilisés de façon performative dans le langage naturel; l’interprétation sémantique actuelle de ces commandes n’est pas satisfaisante.

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

Nous pensons qu’il serait plus approprié de réviser le langage KQML et ACL en utilisant une sémantique faisant appel à la recursion des actes du langage, telle que proposée par Chaib-draa et Vanderveken [28], et un cadre théorique d’interprétation des actes de discours tels que proposé par Vongkasem [141].

Nous avons fait qu’effleurer les communication et le lecteur intéressé par plus de détails, en particulier en ce qui concerne KQML et ACL, pourrait se référer au chapitre du présent volume dédié à la communication entre agents [82].

7. Méthodes de conception des systèmes multiagents

Les SMAs sont souvent des systèmes complexes qui demandent de longs efforts de développement. Du début des années 80 jusqu'au milieu des années 90, les efforts des chercheurs se sont surtout concentrés sur l'exploration de nouveaux concepts, la mise en place de théories, la réalisation de divers prototypes de SMAs. Ce bouillonnement d'idées est tout à fait normal dans un jeune domaine technologique: c'est ce que l'on a observé pour les bases de données dans les années 70, les systèmes experts et les systèmes orientés-objets dans les années 80. Cependant, lorsque la technologie atteint plus de maturité et de stabilité, il devient nécessaire de développer des méthodes de conception qui permettent aux entreprises de développer de façon systématique des systèmes opérationnels dans un environnement industriel. Telle est l'évolution que l'on observe pour les SMAs depuis quelques années. Parmi les travaux des pionniers, citons Decker et ses collègues [43] qui ont proposé un cadre de référence pour évaluer les recherches en résolution coopérative de problèmes distribués ("cooperative distributed problem solving"). Leur but était de comprendre les relations entre des efforts de recherche disparates et d'évaluer les progrès dans ce champ de recherche. Bien que ce cadre ne soit pas une méthode de conception, il soulignait un besoin de prendre du recul par rapport au foisonnement des recherches.

Werner[143] proposa lui aussi un cadre de référence pour essayer de comprendre les "relations qui peuvent s'établir entre les usagers, les développeurs et le SMA en construction". Cette réflexion essayait de situer le développement traditionnel de systèmes informatiques par rapport au développement d'un SMA dans lequel les agents pourraient évoluer par eux-mêmes, choisissant leur propres buts, créant leurs propres plans, etc. Ce cadre n'était pas une méthode de conception à proprement parlé, mais dénotait une préoccupation méthodologique intéressante. Goodson et Schmidt [63] ont proposé la première démarche méthodologique visant à décomposer un problème multiagents en des unités fonctionnelles, d'identifier les unités fonctionnelles que la machine puisse traiter mieux que les usagers et de construire les modules de résolution de problèmes pour ces unités. Moulin et Cloutier [101] ont développé, quant à eux, la méthode MASB ("Multi-Agent Scenario-Based method"), une méthode de conception de SMAs qui se base sur l'analyse et la conception de scénarios caractérisant les rôles joués par les agents humains et les agents artificiels et qui s'applique bien pour des applications de "travail collaboratif" (ex. système distribué de prise de rendez vous).

Burmeister [20] a proposé une méthode dans laquelle on utilise trois modèles pour analyser un SMA : un modèle d'agent (les agents et leur structure interne définie en termes de croyances, buts, plans, etc.); un modèle organisationnel (description des relations entre agents); un modèle de coopération (description des interactions entre agents). Cette méthode empruntait des techniques de modélisation aux méthodes orientées-objet telles que les cartes CRC et la notation d'OMT. Kinny et ses collègues [80] ont proposé une méthode qui distingue deux niveaux de conception (interne et externe) pour la création d'agents BDI. Les agents BDI mettent en oeuvre un modèle basé sur les notions de croyance (Belief), désir (Desire) et intention (Intention). Ce modèle a été popularisé par Rao et Georgeff [110,111,112]. Le niveau externe de ce modèle est spécifié par un modèle qui décrit les relations hiérarchiques entre agents et un modèle d'interaction qui définit les responsabilités, services et interactions entre les agents et leur environnement. Le niveau interne permet de modéliser chaque agent BDI grâce à trois modèles : un modèle de croyances, un modèle de buts et un modèle de plans.

Kendall et ses collègues [79] ont proposé, quant à eux, une approche de conception de SMA qui combine des techniques orientées-objet et des techniques de modélisation d'entreprise et de fabrication assistée par ordinateur. Dans cette méthode, les auteurs ont essayé d'adapter les techniques de modélisation des systèmes orientés-objets au domaine des systèmes multiagents (use cases, diagrammes de scénarios, patrons de conception, etc.). De leur côté, Maamar et Moulin [94] ont proposé une méthode de conception de systèmes composés d'agents hétérogènes en proposant l'utilisation de cadres ("frameworks") orientés agents. L'idée est de spécifier les diverses composantes de systèmes distribués et hétérogènes sous la forme d'équipes d'agents dans lesquelles chaque agent joue des rôles spécifiques. Les frameworks, les équipes et les agents sont caractérisés par les services qu'ils rendent et ceux qu'ils requièrent. Diverses techniques de modélisation sont employées pour spécifier les scénarios, les diagrammes d'interactions des agents, les modèles d'agents. Cette méthode a été utilisée pour la réalisation d'un prototype permettant de réaliser l'interopérabilité de géorépertoires.

La méthode Cassiopea [35] distingue 3 phases pour la conception d'un SMA et a été appliquée au domaine des "robots footballeurs". Cette méthode utilise des techniques orientées-objet pour spécifier les comportements élémentaires des agents. Les comportements relationnels entre agents sont par la suite analysés grâce à un graphe de couplages. Enfin on spécifie la dynamique de l'organisation des agents (qui s'allie avec qui) en utilisant le graphe de couplages. Verharen [138] a, quant à lui, proposé une méthode de conception qui s'appuie sur une perspective de gestion des activités d'un SMA et met notamment en oeuvre les modèles suivants: un modèle d'autorisation qui décrit les types de communications permises et les obligations qui existent entre une organisation et son environnement; un modèle de communication qui raffine le modèle précédent sous la forme de contrats entre les agents sous la forme de réseaux de Pétri; les transactions entre les agents sont modélisées par des diagrammes de transactions qui associent actes de discours et buts des agents; un modèle de tâches qui fournit la décomposition des tâches à partir des diagrammes précédents.

Les méthodes CoMoMAS [65] et MAS-CommonKADS [72] proposent des extensions à la méthode CommonKADS [138] qui a été développée pour le domaine de l'ingénierie des connaissances et en s'appuyant sur certaines techniques orientées-objets (use cases, scénarios, modèles de classes). Ces méthodes définissent divers modèles : le modèle d'agent qui décrit les caractéristiques des agents (capacités de raisonnement, buts, services); le modèle de tâches (tâches et sous-tâches); le modèle d'expertise (connaissances nécessaires aux agents pour réaliser leurs tâches); le modèle de coordination (description des interactions possibles entre agents, protocoles et capacités nécessaires); le modèle d'organisation qui décrit comment le SMA va être introduit dans l'organisation qui accueille la société d'agents; le modèle de communication qui décrit les interactions entre les usagers et les agents; le modèle de conception qui synthétise les modèles précédents en tenant compte des contraintes d'implantation. Ces méthodes ont été utilisées dans divers projets de recherche.

Comme on peut le constater plusieurs méthodes d'analyse et de conception des SMA ont été proposées récemment. La plupart d'entre elles, s'appuient sur des techniques de modélisation empruntées à des méthodes connues en développement orienté-objets ou en ingénierie des connaissances pour aider à la construction des modèles d'agents, de l'architecture du SMA, pour la spécification des modèles d'organisation, d'interaction, etc. Peut-on s'attendre à la création d'une approche standard comme cela fut le cas en ingénierie des connaissances avec la méthode

KADS ou à une normalisation des modèles comme ce fut le cas en orienté-objets avec le langage UML? L'avenir nous le dira. Une telle standardisation apparaîtra certainement quand les technologies multiagents seront adoptées effectivement par les entreprises.

Précisons pour finir, que Guessoum dans le chapitre ?? [68] revient en détail sur les environnements de développement des systèmes multiagents. L'auteure précise ce qu'on entend par « opérationnalisation » des systèmes multiagents, donne quelques outils génériques pour développer des SMAs. Suit alors toute une section dédiée aux bibliothèques SMA pour langages classiques, les langages SMA dédiés et les outils intégrés. Dans sa dernière partie, l'auteur expose son point de vue sur les AGL multiagents.

8. Quelques exemples d'applications des SMA

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

Dans ce qui va suivre nous n'en exposerons que quelques exemples d'applications utilisant cette technologie et nous référons le lecteur à l'article de Jennings [74] pour un ensemble d'exemples plus complet.

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

8.1 Application des SMAs aux télécommunications

Ces dernières années, les télécommunications ont notamment introduit une conception de services décentralisée dans le contexte du Web, créé de nouveaux services de médiation tels que les portails et engendrés l'apparition de nombreux fournisseurs de services réseaux qui ne disposent pas de leurs propres services réseaux [12]. L'obtention de tels services décentralisés ne peut, bien entendu, être obtenue que grâce à des logiciels pour lesquels les données et le contrôle sont forcément distribués. De ce fait, il est clair que les SMAs semblent convenir aux télécommunications. C'est pourquoi les principaux acteurs de télécommunications mènent actuellement d'intenses activités de recherche sur la technologie agent : British Telecom, France Télécom, Deutch Telekom, NTT, Nortel, Siemens, etc. Bourron dans le Chapitre 10 revient en détail sur l'application des SMAs aux télécommunications [12]. Il dresse tout d'abord les orientations de recherche dans les télécommunications et présente ensuite, à travers des exemples et des scénarios, un état de l'art des réalisations en vue de préciser les éléments techniques éprouvés et les problèmes résidants. Au fil de son analyse, Bourron montre le rôle que pourraient jouer les techniques agent dans la réalisation et l'exploitation de services de télécommunication.

8.2 Le système ADEPT

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

Le système ADEPT [5] attaque ce problème en voyant le processus d'affaires comme un ensemble d'agents qui négocient et qui offrent des services. Chaque agent représente un rôle distinct ou un département de l'entreprise et est en mesure de fournir un ou plusieurs services. Les agents qui requièrent les services d'autres agents le font par une négociation qui permet d'obtenir un coût, un délai temporel et un degré de qualité qui sont acceptables aux deux parties. Le résultat d'une négociation terminée avec succès constitue un engagement entre les deux parties.

8.3 Le système GUARDIAN

Le système GUARDIAN [118] a pour but de gérer les soins aux patients d'une unité de soins intensifs chirurgicale. Les principales motivations de ce système sont: premièrement, le modèle des soins d'un patient dans une unité de soins intensifs est essentiellement celui d'une équipe, où un ensemble d'experts dans des domaines distincts coopèrent pour organiser les soins des patients; deuxièmement, le facteur le plus important pour donner de bons soins aux patients est le partage d'informations entre les membres de l'équipe de soins critiques. Particulièrement, les médecins spécialistes n'ont pas l'opportunité de superviser l'état d'un patient minute par minute; cette tâche revient aux infirmières qui, quant à elles, ne possèdent pas les connaissances nécessaires à l'interprétation des données qu'elles rassemblent.

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

8.4 Les systèmes d'informations coopératifs (SIC)

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

³Traduction du terme anglais *effectors*.

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

Le système multiagent « Warren » pourrait constituer un exemple spécifique de l'utilisation des agents dans ce type d'application. C'est un système d'agents intelligents pour l'aide des usagers dans la gestion des portefeuilles [146]. Ce système combine les données du marché financier, les rapports financiers, les modèles techniques et les rapports analytiques avec les prix courants des actions des compagnies. Toutes ces informations sont déjà disponibles sur le Web; "Warren" ne fait que les intégrer via des agents spécialisés, les agents d'informations et ensuite les présenter aux usagers. Pour ce faire, "Warren" dispose de six agents ressources, deux agents de tâches et un agent utilisateur pour chaque usager. L'agent utilisateur affiche (via le web) les informations financières de son usager, lui permettant de faire des simulations d'achat et de vente des actions. Il affiche également les prix courants des actions et les nouvelles informations du marché financier. Le même agent permet également d'accéder aux rapports produits par les deux agents de tâches. Ces deux agents fournissent d'une part, une intégration graphique des prix et des nouvelles concernant les actions et, d'autre part, une analyse fondamentale des actions en tenant compte de leurs historiques. Les agents d'informations accèdent à différentes sources d'informations, comme les pages Web, les nouvelles de "Clarinet et Dow-Jones", les rapports financiers électroniques de "SEC Edgar" ainsi que d'autres rapports sous un format texte. "Warren" n'est qu'un exemple et il existe actuellement plusieurs autres systèmes qui touchent à ce genre d'application. Parmi ces applications, nous pouvons citer :

- Infosleuth : C'est un système multiagent pour la recherche coopérative d'informations dans des bases de données distribuées. Ce système a été appliqué aux domaines médicaux [107].
- NetSA (pour "Networked Software Agents"): C'est un système proche de Infosleuth et dédié aux environnements riches en informations [40,41].
- UMDL : C'est un système d'informations coopératif pour la recherche des documents dans une librairie digitale [139].

À titre d'exemple, le système NETSA est un système multiagent coopératif, développé à l'université Laval [40,41] est destiné aux environnements riches en informations. Ce système comporte plusieurs types d'agents:

- Un agent *utilisateur* en charge de la cueillette et du filtrage des informations provenant et allant vers l'usager;

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

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

1. de pousser l'ingénierie de la construction des systèmes à agents collaboratifs. Il faut concevoir plus de méthodes et d'outils pour faciliter leurs implantations [127];
2. de maîtriser et de bien identifier la coordination entre les agents. Il s'agit en particulier d'établir une théorie claire et formelle pour cette coordination [117];
3. d'assurer les critères de stabilité, de clarté et de performance pour de tels systèmes;
4. de trouver des techniques qui permettent l'évaluation, la vérification et la validation de ces systèmes.

Il est à signaler que NetSA (Networked Software Agents), l'architecture sur laquelle nous avons travaillé, fait elle aussi partie des systèmes d'informations coopératifs.

9. Conclusion

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

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

Une des forces importantes derrière la croissance rapide que connaissent aujourd'hui les systèmes multiagents est l'Internet, où la population d'agents est sans cesse croissante. Ces agents devront savoir collaborer afin d'atteindre les objectifs de leurs concepteurs. Dans ce type d'environnement, les agents rencontrent deux défis majeurs: ils doivent être en mesure de se rencontrer et inter-opérer. Une première solution à ce problème est l'introduction d'agents intermédiaires (*brokers, facilitateurs, etc...*) [40,41,135]. Ces agents intermédiaires ont pour principales fonctions : (1) d'associer au mieux les besoins des utilisateurs et les services des fournisseurs; (2) d'unifier et probablement traiter les réponses des fournisseurs pour produire un résultat approprié; (3) aviser régulièrement les utilisateurs du changement des informations.

Dans cette optique d'inter-opérabilité, un grand nombre de langages de communications pour les agents ont été développés; basés, pour plupart, sur la théorie des actes du discours. Bien que les performatives offertes par ces langages permettent de caractériser les types des messages, ils ne permettent pas encore aux agents de comprendre explicitement les concepts "discutés". Le problème des ontologies reste donc ouvert.

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

Finalement, il convient de préciser que les chercheurs travaillant sur les SMAs d'un point de vue formel se sont presque toujours heurtés à des agents omniscients, c'est à dire des agents ayant des capacités de raisonnement illimitées. Là aussi le problème est très ouvert et il y a présentement pas mal d'équipes qui y travaillent.

Comme on peut encore le constater le domaine des systèmes multiagents demeure encore aujourd'hui un domaine rempli de défis à surmonter, autrement dit un domaine très ouvert pour la recherche.

Références

- [1] G. Agha and C. Hewitt. Concurrent programming using actors. In A. Yonezawa and M. Tokoro, editors, *Object-Oriented Concurrent Programming*, Computer Systems Series, pages 37-53. MIT Press, 1988.
- [2] G. Agha. *Actors : A Model Of Concurrent Computation In Distributed Systems*. MIT Press, Cambridge, MA, 1986.
- [3] J. Allen, J. Hendler, and A. Tate. *Readings in Planning*. Representation and Reasoning Series. Morgan Kaufmann, San Mateo, California, 1990.

- [4] J. F. Allen and C. R. Perrault. Analyzing intention in dialogues. *Artificial Intelligence*, 15(3), 1980.
- [5] J. A. Alty, D. Griffiths, N. R. Jennings, E. H. Mamdani, A. Struthers, and M. E. Wiegand. ADEPT : Advanced decision environment for process tasks : Overview & architecture. In *Proceedings of the 1994 BCS Expert Systems Conference (Applications Track, ISIP Theme)*, pages 359-371, Cambridge, UK, 1994.
- [6] D. E. Appelt. *Planning English Sentences*. Cambridge University Press : Cambridge, England, 1985.
- [7] G. Babin, Z. Maamar, and B. Chaib-draa. Metadatabase meets distributed AI. In *First Int. Workshop on Cooperative Information Agents (CIA-97)*, 1997.
- [8] M. Barbuceanu M. and Fox M.S. Cool : a language for describing coordination in multiagent systems. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-1)*, pages 17-24, San Francisco, CA, 1995.
- [9] J. Bates. The role of emotion in believable agents. *Communications of the ACM*, 37(7): 122-125, July 1994.
- [10] J. Bates, A. B. Loyall, and W. S. Reilly. An architecture for action, emotion, and social behaviour. Technical Report CMU-CS-92-144, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, May 1992.
- [11] A. H. Bond and L. Gasser, editors. *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers : San Mateo, CA, 1988.
- [12] T. Bourron. Application des systèmes multiagents dans les télécommunications : États de l'art, enjeux et perspectives. *Le présent volume*.
- [13] K. Bouzouba and B. Moulin. Le projet postage : Vers l'amélioration des communications entre agents. *Information In Cognito*, 11-12 :27-43, 1998.
- [14] M. Bratman. *Intention, plans, and practical reason*. Harvard University Press, 1987.
- [15] M. Bratman, D. Israel, and M. Pollack. Plans and resource bounded practical reasoning. *Computational Intelligence*, 4 :349-355, 1988.
- [16] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, RA-2(1) :14-23, April 1986.
- [17] R. A. Brooks. Intelligence without representation. *Artificial Intelligence*, 47(1-3) :139-159, January 1991.

- [18] R. A. Brooks. Intelligence without reason. In Ray Myopoulos, John ; Reiter, editor, *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pages 569-595, . Morgan Kaufmann, Sydney, Australia, 1991.
- [19] F. Bousquet and C. Le Page. Systèmes multi-agents et écosystèmes. Le présent volume.
- [20] B. Burmeister. Models and methodology for agent-oriented analysis and design. In K. Fisher, editor, *Working Notes on the KI'96 Workshop on Agent-Oriented Programming and Distributed Systems*, pages 187-223. DFKI Document, 1996.
- [21] Castelfranchi C. and Conte R. Distributed artificial intelligence and social science : critical issues. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*, pages 527-556. John Wiley & Sons : Chichester, England, 1996.
- [22] S. Cammarata, D. McArthur, and R. Steeb. Strategies of cooperation in distributed problem solving. *Proceedings of the 8th International Joint Conference on Artificial Intelligence*, pages 767-770, 1983. (Also published in *Readings in Distributed Artificial Intelligence*, Alan H. Bond and Les Gasser, editors, pages 102-105, Morgan Kaufmann, 1988).
- [23] S. Cammarata, D. McArthur, and R. Steeb. Strategies of cooperation in distributed problem solving. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)*, Karlsruhe, Germany, 1983.
- [24] B. Chaib-draa and P. Levesque. Hierarchical models and communication in multi-agent environments. In *Proceedings of the Sixth European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-94)*, pages 119-134, Odense, Denmark, August 1994.
- [25] B. Chaib-draa. Distributed Artificial Intelligence : An overview. In A. Ken, J. G. Williams, C. M. Hall, and R.Kent, editors, *Encyclopedia Of Computer Science And Technology*, volume 31, pages 215-243. Marcel Dekker, Inc, 1994.
- [26] B. Chaib-draa. Industrial applications of distributed AI. *Communications of the ACM*, 38(11) :49-53, 1995. [25] B. Chaib-draa. Interaction between agents in routine, familiar and unfamiliar situations. *International Journal of Intelligent and Cooperative Information Systems*, 1(5):7-20, 1996.
- [27] B. Chaib-draa and P. Levesque. Hierarchical model and communication by signs, signals and symbols in multiagent environments. *Journal of Experimental and Theoretical AI (JETAI)*, 8 :7-20, 1996.
- [28] B. Chaib-draa, and D. Vanderveken. Agent communication language : towards a semantics based on success, satisfaction and recursion, in *Intelligent Agents V, Agent Theories, Architectures and Language*, Müller, J.P., Singh M.P., Rao A.S. (Eds), Lecture Notes on AI, Springer Verlag, Berlin, 1999.

- [29] D. Chapman and P. Agre. Abstract reasoning as emergent from concrete activity. In M. P. George- and A. L. Lansky, editors, *Reasoning About Actions & Plans - Proceedings of the 1986 Workshop*, pages 411-424. Morgan Kaufmann Publishers : San Mateo, CA, 1986.
- [30] A. Chavez and P. Maes. Kasbah : An agent marketplace for buying and selling goods. In *First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pages 75-90, 1996.
- [31] P. R. Cohen and C. R. Perrault. Elements of a plan based theory of speech acts. *Cognitive Science*, 3 :177-212, 1979.
- [32] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42 :213-261, 1990.
- [33] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 221-256. The MIT Press : Cambridge, MA, 1990.
- [34] P. R. Cohen and H. J. Levesque. Communicative actions for artificial agents. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS- 1)*, pages 65-72, San Francisco, CA, 1995.
- [35] A. Collinot, A. Drogoul, and P. Benhamou. Agent-oriented design of a soccer robot team. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS- 2)*, pages 41-47, Kyoto, Japan, 1996.
- [36] S. E. Conry, K. Kuwabara, V. R. Lesser, and R. A. Meyer. Multistage negotiation in distributed constraint satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics*, December 1991.
- [37] S. E. Conry, R. A. Meyer, and R. P. Pope. Mechanisms for assessing the nonlocal impact of local decisions in distributed planning. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence*. Volume II, pages 245-258. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989.
- [38] R. Conte, M. Miceli, and C. Castelfranchi. Limits and levels of cooperation. In Y. Demazeau and J.-P. Müller, editors, *Decentralized AI 2 - Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, pages 147-160. Elsevier Science Publishers B.V. : Amsterdam, The Netherlands, 1991.
- [39] D. D. Corkill and V. R. Lesser. The use of meta-level control for coordination in a distributed problem solving network. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 748-756, Karlsruhe, Federal Republic of Germany, August 1983.

- [40] M. Côté and N. Troudi. Une architecture multiagent pour la recherche. *L'expertise Informatique*, 3(3), 1998.
- [41] M. Côté. *Une architecture multiagent et son application aux systèmes financiers*. Master's thesis, Département d'Informatique, Université Laval, Janvier 1999.
- [42] R. Davis and R. Smith. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20(1) :63-109, January 1983.
- [43] K. S. Decker, E. H. Durfee, and V. R. Lesser. Evaluating research in cooperative distributed problem solving. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence. Volume II*, pages 487-519. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [44] Y. Demazeau and J.-P. Müller, editors. *Decentralized AI 2 - Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*. Elsevier Science Publishers B.V. : Amsterdam, The Netherlands, 1991.
- [45] E. H. Durfee and V. R. Lesser. Using partial global plans to coordinate distributed problem solvers. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, Milan, Italy, 1987.
- [46] E. H. Durfee. *Coordination of distributed problem solvers*. The Kluwer international series in engineering and computer science ; Artificial intelligence. Kluwer, Boston, 1988.
- [47] E. H. Durfee and V. Lesser. Negotiating task decomposition and allocation using partial global planning. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 229-244. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [48] R. S. Englemore and T. Morgan. *Blackboard Systems*. Addison-Wesley, 1989.
- [49] El-Fallah Seghrouni A. Modèles de coordination d'agents cognitifs. *Le présent volume*.
- [50] El-Fallah Seghrouni, A. and Haddad, S. A coordination algorithm for multiagent planning. In *Proc. of Int. Conf. on Multiagent Systems-1996 (ICMAS'96)*, AAAI, Press, Kyoto, Japan, 1996.
- [51] El-Fallah Seghrouni, A. and Haddad, S. A formal model for coordinating plans in multiagent systems. In *Proc. of Intelligent Agents Workshop*, Augusta Technology Ltd, Brooks University, Oxford UK, 1995.
- [52] R. Fagin, I. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge*. the MIT Press : Cambridge, MA, 1995.
- [53] J. Ferber. *Les systèmes multi-agents, vers une intelligence collective*. InterEditions, 1995.

- [54] I. A. Ferguson. *TouringMachines : An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, Clare Hall, University of Cambridge, UK, November 1992. (Also available as Technical Report No. 273, University of Cambridge Computer Laboratory).
- [55] R. Fikes and Nils Nilsson. strips : A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3/4) :189-208, 1971.
- [56] T. Finin and R. Fritzson. KQML: a language and protocol for knowledge and information exchange. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence*, pages 126-136, Lake Quinalt, WA, July 1994.
- [57] K. Fischer, B. Chaib-draa, H. J. Müller, J. P. Müller, and M. Pischel. A simulation approach based on negotiation and cooperation between agents. *IEEE Trans. on Systems, Man, and Cybernetics*, 29(4), 1999, pp. 531-545.
- [58] S. Franklin and A. Graesser. Is it an agent, or just a program ? : A taxonomy for autonomous agents. In J. P. Mueller, M. Wooldridge, and N. R. Jennings, editors, *Intelligent Agents III : Theories, Architectures, and Languages (LNAI Volume 1193)*, pages 21-35. Springer-Verlag : Heidelberg, Germany, 1997.
- [59] J. R. Galliers. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK, 1988.
- [60] J. R. Galliers. Cooperative interaction as strategic belief revision. In S. M. Deen, editor, *CKBS-90 - Proceedings of the International Working Conference on Cooperating Knowledge Based Systems*, pages 148-163. Springer-Verlag : Heidelberg, Germany, 1991.
- [61] L. Gasser. Social conceptions of knowledge and action : DAI foundations and open systems semantics. *Artificial Intelligence*, 47 :107-138, 1991.
- [62] M. P. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, Washington, D.C., 1983.
- [63] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *The Proceedings of AAAI-87*, pages 677-682, Seattle, 1987.
- [64] M. Georgeff. Communication and interaction in multiagent planning. In Bond A.H. and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 200-204. Morgan Kaufman, 1998.
- [65] N. Glaser. *Contributing to Knowledge Modeling in a Multi-Agent Framework : the CoMoMAS Approach*. PhD thesis, Université Henri Poincaré, Nancy 1, France, 1996.
- [66] J. L. Goodson and C. F. Schmidt. The design of cooperative person-machine problem-solving systems : A methodology and an example. In W. W. Zachary and S. P. Robertson, editors, *Cognition, Computation and Cooperation*, pages 187-223. Ablex, 1990.

- [67] R. Gray. Agent Tcl : A flexible and secure mobile-agent systems. Technical Report PCS-TR98-327, Dartmouth College, Computer Science, Hanover, NH, January 1998.
- [68] Z. Guessoum. Environnements de développement des systèmes multiagents. Le présent volume.
- [69] J. Y. Halpern. Reasoning about knowledge : A survey. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 1-34. Oxford University Press : Oxford, England, 1995.
- [70] B. Hayes-Roth, G. Ball, G. Lisette, R. Picard, and A. Stern. Affect and emotion in the user interface. In *Proceedings of the International Conference on Intelligent User Interface (IUI-98)*. ACM Press, 1998.
- [71] M. Huhns, U. Mukhopadhyay, and L. M. Stephens. DAI for document retrieval : The MINDS project. In M. Huhns, editor, *Distributed Artificial Intelligence*, pages 249-284. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1987.
- [72] C.A. Iglesias, M. Garijo, J.C. González, and J.R. Velasco. Analysis and design of multiagent systems using mas-commonkads. In *AAAI'97 Workshop on Agent Theories, Architectures and Languages*, Providence, RI, 1997.
- [73] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 74(2), 1995.
- [74] N. R. Jennings, M. Wooldridge, and K. Sycara. A roadmap of agent research and development. *Int Journal of Autonomous Agents and Multi-Agent Systems*, 1(1) :7- 38, 1998.
- [75] N.R. Jennings and M.J. (eds) Wooldridge. *Agent Technology : Foundations, applications, and Markets*. Springer-Verlag : Heidelberg, Germany, 1998.
- [76] N. R. Jennings. Agent methodology for software engineering. *Communication of ACM*, (to appear), 2000.
- [77] L. P. Kaelbling. An architecture for intelligent reactive systems. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning About Actions & Plans - Proceedings of the 1986 Workshop*, pages 395-410. Morgan Kaufmann Publishers : San Mateo, CA, 1986.
- [78] F. Kaplan. A new approach to class formation in multi-agent simulation of language evolution. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-2)*, pages 158-165, Paris, Fr, 1998.
- [79] E. A. Kendall, M. T. Malkoun, and C. Jiang. A methodology for developing agent-based systems for enterprise integration. In D. Lukose and Zhang C., editors, *First Australian Workshop on Distributed Artificial Intelligence*, pages 216 - 231. Springer Verlag, LNAI 1087, 1996.

- [80] D. Kinny, M. Georgeff, and A. Rao. Methodology and modelling technique for systems of bdi agents. In W.van der Welde and J. Perram, editors, *Decentralized AI 2 - Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*. Springer Verlag Lecture Notes in AI 1038, 1996.
- [81] A. Kobsa. *User Models in Dialog Systems*. Springer-Verlag : Heidelberg, Germany, 1989.
- [82] J. L. Koning and Presty, S. Modèles de communication. Le present volume.
- [83] S. Kraus and J. Wilkenfeld. Negotiations over time in a multi-agent environment. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 56-61, Sydney, Australia, 1991.
- [84] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2) :297-345, 1995.
- [85] T. Kreifelts and F. von Martial. A negotiation framework for autonomous agents. In Y. Demazeau and J.-P. Müller, editors, *Decentralized AI 2 - Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, pages 71-88. Elsevier Science Publishers B.V. : Amsterdam, The Netherlands, 1991.
- [86] G. Lakemeyer and H. Levesque. Query evaluation and progression in AOL knowledge bases. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 124-131, Stockholm, Sweden, 1999.
- [87] Y. Lespérance. *A Formal Theory of Indexical Knowledge and Action*. PhD thesis, Toronto, 1991.
- [88] Y. Lespérance, H. J. Levesque, F. Lin, D. Marcu, R. Reiter, and R. B. Scherl. A logical approach to high-level programming- A progress report. In Benjamin Kuipers, editor, *In Control of the Physical World by Intelligent Systems, Papers from the 1994 AAAI fall Symposium*, pages 78-85, New Orleans, LA, 1994.
- [89] V. R. Lesser and L. D. Erman. Distributed interpretation : A model and experiment. *IEEE Transactions on Computers*, C-29(12), 1980.
- [90] V. R. Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 21(6) :1347-1362, 1991.
- [91] V. R. Lesser. A retrospective view of FA/C distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 21 :1347-1363, 1991.

- [92] H. J. Levesque, J. H. T. Nunes, and P. R. Cohen. On acting together. In William Dietterich, Tom ; Swartout, editor, *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 94-99, Hynes Convention Centre ?, 1990. MIT Press.
- [93] D. J. Litman and P. R. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11 :163-200, 1987.
- [94] Z. Maamar and B. Moulin. An agent based approach for intelligent and cooperative systems. In *Proc. of IEEE Knowledge and Data Engineering Exchange Workshop'97 (KDEX-97)*, pages 41-47, Newport Beach, California, 1997.
- [95] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, 37(7) :31-40, July 1994.
- [96] P. Maes. Social interface agents : Acquiring competence by learning from users and other agents. In O. Etzioni, editor, *Software Agents - Papers from the 1994 Spring Symposium (Technical Report SS-94-03)*, pages 71-78. AAAI Press, March 1994.
- [97] T. W. Malone. Organizing information processing systems : parallels between human organizations and computer systems. In W. W. Zachary and S. P. Robertson, editors, *Cognition, Computation and Cooperation*, pages 56-83. Ablex, 1990.
- [98] E. Malville and F. Bourdon. Task allocation : A group self-design approach. In *Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS- 2)*, pages 166-167, Kyoto, Japan, 1998.
- [99] H. Mintzberg. *The Structuring of Organizations*. Englewoods Cliffs, 1979.
- [100] B. Moulin and B. Chaib-draa. An overview of distributed artificial intelligence. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*, pages 3-54. John Wiley & Sons : Chichester, England, 1996.
- [101] B. Moulin and L. Cloutier. Collaborative work based on multiagent architectures : a methodological perspective. In F. Aminzadeh and M. Jamshidi, editors, *Artificial Intelligence : Theory and Applications*, pages 187-223. Prentice Hall, 1994.
- [102] J. Müller. Negotiation principles. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*, pages 211-230. John Wiley & Sons : Chichester, England, 1996.
- [103] A. Negroponte. *Being Digital*. Hodder and Stoughton, 1995.
- [104] A. Newell and H. A. Simon. GPS, a program that simulates human thought. In Heinz Billing, editor, *Lernende Automaten*, pages 109-124. R. Oldenbourg, Munich, Germany, 1961.

- [105] P. Nii. Blackboard systems. In *Handbook of Artificial Intelligence*. Volume IV. Addison-Wesley : Reading, MA, 1989.
- [106] P. Nii, N. Aiello, and J. Rice. Experiments on Cage and Poligon : Measuring the performance of parallel blackboard systems. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence*. Volume II, pages 319-384. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [107] M. Nodin, W. Bohrer, and A. Hiong Ngu. Semantic brokering over dynamic heterogeneous data sources in infosleuth. Technical Report ??, MCC Corporation, 1998.
- [108] H.S. Nwana and D. T. Ndumu. Agents of change in future communication systems. *Applied AI Journal*, 13(1), 1999.
- [109] H. V. D. Parunak. Applications of distributed artificial intelligence in industry. In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*. John Wiley & Sons: Chichester, England, 1996.
- [110] A. S. Rao and M. P. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 498-504, Sydney, Australia, 1991.
- [111] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473-484. Morgan Kaufmann Publishers : San Mateo, CA, April 1991.
- [112] A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In William Nebel, Bernhard ; Rich, Charles ; Swartout, editor, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 439-449, Cambridge, MA, 1992. Morgan Kaufmann.
- [113] J. S. Rosenschein, M. Ginsberg, and M. R. Genesereth. Cooperation without communication. In *Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86)*, Philadelphia, PA, 1986.
- [114] S. Rosenschein. Formal theories of knowledge in AI and robotics. *New Generation Computing*, pages 345-357, 1985.
- [115] S. J. Rosenschein. Rational Interaction : *Cooperation Among Intelligent Agents*. PhD thesis, Stanford University, Computer Science Department, 1985.
- [116] S. J. Rosenschein and J. S. Breese. Communication-free interactions among national agents. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 99-118. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.

- [117] S. J. Rosenschein and G. Zlotkin. *Rules of Encounter : Designing Conventions for Automated Negotiation Among Computer*. MIT Press, Boston, MA, 1994.
- [118] B. H. Roth, M. Hewett, R. Washington, R. Hewett, and A. Seiver. Distributing intelligence within an individual. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence*, volume 2 of Research Notes in Artificial Intelligence, pages 385-412. Pitman, 1989.
- [119] D. Rousseau and G. Moulin, B. and Lapalme. Cool : a language for describing coordination in multiagent systems. In *Actes des Deuxièmes journées francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, pages 3-14, Voiron, (France), 1995.
- [120] D. Rus and D. Subramanian. Information retrieval, information structure, and information agents. Technical Report PCS-TR96-255, Dartmouth College, Computer Science, Hanover, NH, January 1996.
- [121] T. Sandholm and V. Lesser. Issues in automated negotiation and electronic commerce : Extending the contract net framework. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-1)*, pages 328-335, San Francisco, CA, 1995.
- [122] J. R. Searle. *Speech Acts : An Essay in the Philosophy of Language*. Cambridge University Press : Cambridge, England, 1969.
- [123] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60(1) :51-92, 1993.
- [124] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Diego, CA, 1992.
- [125] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, 60 :51-92, 1993.
- [126] M. P. Singh. *Multiagent Systems : A Theoretical Framework for Intentions, Know-How, and Communications (LNAI Volume 799)*. Springer-Verlag : Heidelberg, Germany, 1994.
- [127] D. C. Smith, A. Cypher, and J. Spohrer. KIDSIM : Programming agents without a programming language. *Communications of the ACM*, 37(7) :55-67, July 1994.
- [128] R. G. Smith. The contract net protocol. *IEEE Transactions on Computers*, C-29(12), 1980.
- [129] R. G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1), 1980.
- [130] K. P. Sycara. Resolving goal conflicts via negotiation. In *Proceedings of the 7th National Conference on Artificial Intelligence*, pages 245-250, St. Paul, Minnesota, 1988.

- [131] K. P. Sycara. Multiagent compromise via negotiation. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 119-138. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [132] K. P. Sycara. Negotiation planning : An AI approach. *European Journal of Operational Research*, 46 :216-234, 1990.
- [133] K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28 :203-242, 1990.
- [134] S. R. Thomas. *PLACA, an Agent Oriented Programming Language*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA 94305, August 1993. (Available as technical report STAN-CS-93-1487).
- [135] N. Troudi. Un système multiagent pour les environnements riches en information. Master's thesis, Département d'Informatique, Université Laval, Avril 1999.
- [136] S. L. (ed.) Tubbs. *A system Approach to Small Group Interaction*. 2nd edition, Addison Wesley, Reading MA, 1984.
- [137] F. V. Martial, editor. *Coordinating Plans of Autonomous Agents*. Springer-Verlag : Heidelberg, Germany, 1992.
- [138] E. M. Verharen. E. M. *A Language-Action Perspective on the Design of Cooperative Information Agents*. PhD thesis, Katholieke Universiteit Brabant, The Netherlands, 1997.
- [139] M.J. Vidal, T. Müller, P. Weinstein, and Durfee H. E. The university of michigan digital library service market society. In K. P. Sycara and M. Wooldridge, editors, *Proc. of the 2nd Int. Conf. on Autonomous Agent (Agent-98)*, pages 269-276, 1998.
- [140] F. von Martial. Interactions among autonomous planning agents. In Y. Demazeau and J.-P. Müller, editors, *Decentralized AI - Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*. Elsevier Science Publishers B.V. : Amsterdam, The Netherlands, 1990.
- [141] L Vongkasem and B. Chaib-draa. Acl as a joint project between participants : A preliminary report. In *First Workshop on Agent Communication Language (ACL'99)*, Stockholm, Sweden, 1999.
- [142] P. Wavish and M. Graham. Roles, skills, and behaviour : a situated action approach to organising systems of interacting agents. In *Intelligent Agents : theories, Architectures, and Languages (LNAI volume 890)*, pages 371-385. Springer-Verlag : Heidelberg, Germany, 1995.

- [143] E. Werner. Cooperating agents : A unified theory of communication and social structure. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence* Volume II, pages 3-36. Pitman Publishing : London and Morgan Kaufmann : San Mateo, CA, 1989.
- [144] E. Werner. The design of multi-agent systems. In E. Werner and Y. Demazeau, editors, *Decentralized AI 3 - Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 3-30. Elsevier Science Publishers B.V. : Amsterdam, The Netherlands, 1992.
- [145] D. E. Wilkins. *Practical Planning : Extending the AI Planning Paradigm*. Morgan Kaufmann, San Mateo, California, 1988.
- [146] D. Zeng and K. Sycara. Coordination of multiple intelligent software agents. *International Journal of Cooperative Information Systems*, 5 :181-212, 1996.