

Chapitre 5

Les Systèmes Multi-Agents

Table des matières

5.1- Introduction	88
5.1.1- Historique	88
5.1.2- Pourquoi distribuer l'intelligence?	88
5.1.3- Qu'est que l'intelligence artificielle distribuée (IAD) ?.....	91
5.1.4- Le monde est ouvert	93
5.1.5- Domaines d'intérêts	93
5.2- Concepts de base.....	93
5.2.1- Agent	93
5.2.1.1- Définitions.....	93
5.2.1.2- Des Objets aux Agents	96
5.2.2- Système Multi-Agents.....	97
5.2.2.1- Qu'est-ce qu'un système multi-agents ?	97
5.2.2.2- Utilité des systèmes multi-agents.....	97
5.2.2.3- Un premier exemple	98
5.2.2.4- Vue intuitive d'un Agent dans un SMA.....	99
5.2.2.5- Variables globales et locales et les SMA	99
5.2.2.6- Niveaux d'organisation	99
5.2.3- Propriétés d'un agent intelligent	100
5.2.3.1- Autonomie.....	100
5.2.3.2- Réactivité.....	100
5.2.3.3- Proactivité	101
5.2.3.4- Adaptabilité	101
5.2.3.5- Sociabilité.....	101
5.2.3.6- Apprentissage	101
5.2.3.7- Sécurité.....	102
5.2.4- Propriétés des systèmes multi-agents.....	102
5.2.4.1- Interactions entre agents.....	102
5.2.4.2- Coopération	103
5.2.4.3- Coordination.....	103
5.2.4.4- La compétition	104

5.2.4.5- Délégation	104
5.2.4.6- Communication	105
5.2.4.7- Une Recherche de Compromis.....	105
5.3- Les différents modèles d'agents (Architecture)	105
5.3.1- Les agents réactifs	107
5.3.1.1- Agents à réflexes simples.....	107
5.3.1.2- Agents conservant une trace du monde.....	108
5.3.2- Les agents délibératifs.....	109
5.3.2.1- Agents ayant des buts.....	110
5.3.2.2- Agents utilisant une fonction d'utilité.....	110
5.3.2.3- Le modèle BDI.....	111
5.3.3- Les agents hybrides	112
5.4- Apprentissage des agents et des SMA.....	113
5.4.1- Apprentissage des Agents	113
5.4.1.1- Définitions et Différentes formes d'apprentissage	113
5.4.1.2- Apprentissage des agents	114
5.4.1.2- L'apprentissage par renforcement.....	116
5.4.2- Apprentissage des SMA.....	117
5.5- Méthodologies de conception d'un SMA.....	117
5.5.1- Problématique	117
5.5.2- Méthodologie	118
5.5.2.1- Phase d'analyse	118
5.5.2.2- Phase de conception	119
5.5.2.3- Les étapes de réalisation d'un SMA	120
5.5.3- Plates-formes de développement.....	120
5.6- Conclusion	121

5.1- Introduction

5.1.1- Historique

Durant la première génération des programmes informatiques, l'ordinateur était chargé de réaliser des tâches prises en charge habituellement par un homme comme par exemple la classification automatique d'une population qui requiert de l'intelligence artificielle. Ce remplacement progressif de l'homme par une machine s'est accompagné d'une identification de la machine à l'humain, un programme représentant l'expert capable de résoudre le problème par lui-même. Cette façon de concevoir les programmes comme des sortes de penseurs repliés sur eux-mêmes a trouvé sa limitation lorsqu'on a cherché à développer des applications plus complexes réalisées habituellement non pas par une seule personne mais par un groupe de personnes parfois délocalisées.

L'informatique devient ainsi de plus en plus diffuse et distribuée dans de multiples objets et fonctionnalités qui sont amenés à coopérer. La décentralisation est donc la règle et une organisation coopérative entre modules logiciels est un besoin. De plus, la taille, la complexité et l'évolutivité croissante de ces nouvelles applications informatiques font, de cette vision centralisée, rigide et passive (contrôlée explicitement par le programmeur), atteindre ses limites.

La machine devait alors être identifiée non plus uniquement à un humain mais à une société organisée d'humains. En particulier, les concepteurs de systèmes industriels complexes ont constaté que le savoir-faire, les compétences et les connaissances diverses sont détenues par des individus différents qui, au sein d'un groupe, communiquent, échangent leurs connaissances et collaborent à la réalisation d'une tâche commune. Les méthodes de réalisation d'applications informatiques se sont alors concentrées sur les aspects organisationnels des logiciels et sur la représentation des communications entre ses différents composants.

Ainsi une nouvelle manière de penser a surgit, donnant naissance à ce qu'on appelle l'intelligence artificielle distribuée.

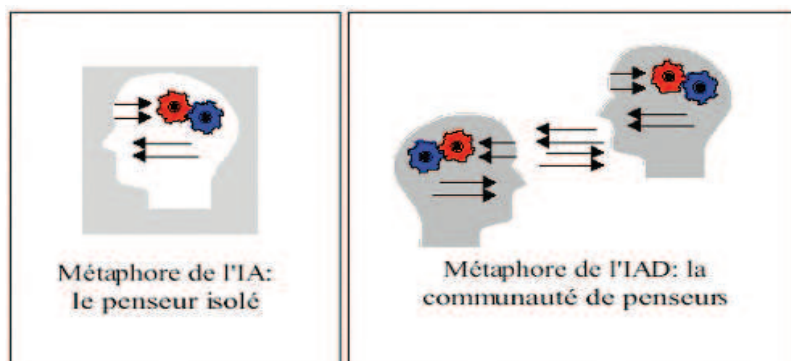


Figure 5.1 : IA versus l'IA.D

Une réponse à la question pourquoi distribuer l'intelligence ? la définition des concepts de base des SMA ainsi que les différents modèles d'agents(Architecture) sont étalés dans la section suivante.

5.1.2- Pourquoi distribuer l'intelligence?

Pourquoi cherche-t-on à créer des intelligences collectives? Pourquoi vouloir à tout prix prendre un point de vue local? Pourquoi tout simplement vouloir distribuer l'intelligence?

L'Informatique « moderne » contient des systèmes de plus en plus répartis parallèlement, qui peuvent donner des éléments de réponse à ces questions :

L'accroissement des capacités informatiques

- Connectivité (internet, WWW, Satellite...)
- puissance de calcul
- Puissance de transmission (vitesse, bande passante...)
- Interface (visualisation, vocal...).
- Informations (taille, complexité, modalité).
- Ressources (distribuées, hétérogènes partagées)

L'accroissement de l'hétérogénéité

- Interactivité (collecticiels, environnement)
- données/connaissances
- BDD multimédia, scientifiques numériques
- Syntaxe, sémantique, structuration
- Régionalisation des informations
- Interface multimodales (données, voix, image, geste, vidéo, langage...).
- Communication (satellite, communication mobiles, réseaux).

L'accroissement des besoins applicatifs

- Traitement intégrés, omniprésents, dynamiques (sujets, information, activité).
- Système d'information hétérogène, étendus, fortement intégrés, très complexes.
- De nouveaux utilisateurs (population globale, communautés d'intérêt, multiples intérêts).
- De nouveaux champs (domaines de connaissance application).
- De nouvelles opportunités d'information, de cout, de transaction, de valorisation.

Les problèmes sont physiquement distribués

Réseaux, contrôle aérien, robotique, etc...

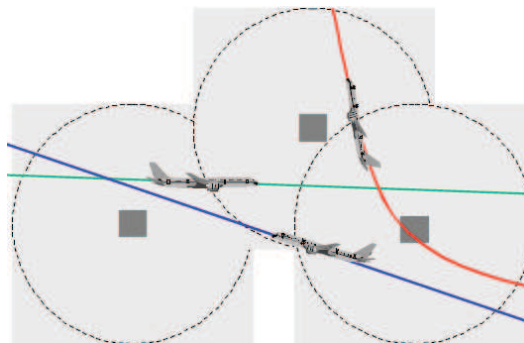


Figure 5.2 : Distribution physique

Les problèmes sont fonctionnellement très distribués et hétérogènes

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

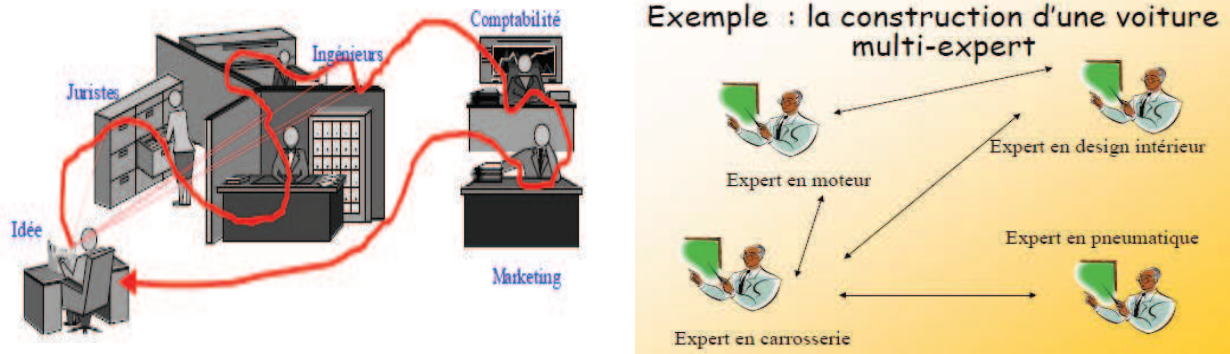


Figure 5.3 : Distribution fonctionnelle

Les réseaux imposent une vision distribuée

A l'heure des réseaux interplanétaires (Internet), où toute l'information et la puissance de traitement est répartie sur un nombre très grand de sites, il faut penser en termes de systèmes ouverts, c'est-à-dire en termes d'interopérabilité radicale des systèmes informatiques. Il faut considérer que l'espace informatique n'est qu'une gigantesque toile d'araignée (WWW) dans laquelle tout ordinateur, qu'il soit fixe, ou mobile, est connecté à l'ensemble des autres ordinateurs du monde entier. Les SMA se présentent ainsi comme des candidats sérieux pour la construction d'architectures ouvertes, distribuées, hétérogènes et souples, capables d'offrir une grande qualité de service dans un travail collectif, sans imposer une structure a priori.

La complexité des problèmes

Impose une vision locale. Lorsque les problèmes sont trop vastes pour être analysés globalement, les solutions fondées sur des approches locales permettent souvent de les résoudre plus rapidement. Par exemple, la régulation de trafic aérien est un problème global complexe, difficile à résoudre du fait du grand nombre de paramètres qui sont mis en jeu et de l'ensemble des contraintes qui doivent être satisfaites, des approches locales permettent de résoudre efficacement ce type de problème.

Les systèmes doivent pouvoir s'adapter à des modifications de structure ou d'environnement

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

Le génie logiciel va dans le sens d'une conception en termes d'unités autonomes en interactions

L'histoire du développement des logiciels montre que la réalisation de programmes informatique suit une démarche visant à la réalisation de systèmes conçus comme des ensembles d'entités de plus en plus distribuées, mettant en jeu des composants davantage individualisés et autonomes. Le récent développement des langages à objets dans tous les secteurs de l'informatique est là pour en témoigner: le génie logiciel passe par la réalisation de

modules autonomes capables d'interagir les uns avec les autres, même, et surtout, lorsque ceux-ci sont conçus par des personnes, des équipes ou des entreprises différentes. Il faut donc associer la fluidité des calculs, la distribution des traitements et l'hétérogénéité des réalisations. Les SMA ont ici un rôle essentiel à jouer en s'inscrivant comme les possibles successeurs des systèmes à objets, en ajoutant à la localité des comportements l'autonomie et la répartition des prises de décision. On peut ainsi déjà parier que le génie logiciel de demain sera "orienté agent", comme celui d'aujourd'hui qui est "orienté objet" (Ferber, 1995).

dans l'espace : il est difficile de décomposer le système en parties déboguables indépendamment ; si on coupe un système en deux on le "tue".

dans le temps : il est difficile d'analyser des systèmes en fonctionnement en continu ; si on arrête un système pour l'observer on le "tue".

Tous ces facteurs peuvent justifier la nécessité de cette distribution d'où des systèmes :

- de plus en plus complexes
- répartis sur des sites de plus en plus nombreux
- constitués de logiciels en interaction entre eux ou avec des êtres humains d'où une volonté
 - D'intégrer :
 - De faire inter opérer :
 - De faire coopérer des logiciels existants.

5.1.3- Qu'est que l'intelligence artificielle distribuée (IAD) ?

Les systèmes experts (qui sont l'émanation la plus visible de l'IA dans l'entreprise) ne peuvent donc prétendre qu'à un rôle mineur dans les procédures de travail informatisé :

- unicité de leur expertise,
- unicité de leur point de vue,
- rigidité de leur capacité d'interaction.

« L'IAD est l'étude, la conception et la réalisation de systèmes multi-agents, c'est à-dire de systèmes dans lesquels des agents intelligents qui interagissent, poursuivent un ensemble de buts ou réalisent un ensemble d'actions » (Wies, 1995).

« L'intelligence, comme la science, n'est pas une caractéristique individuelle que l'on pourrait séparer du contexte social dans lequel elle s'exprime » (Latour, 1989), (Latour, 2006), (Latour & Lemonnier, 1994), (Lestel, 1986), (Lestel & all, 1994).

« Un être humain ne peut se développer convenablement s'il ne se trouve pas entouré d'autres êtres de son espèce. Sans un entourage adéquat, son développement cognitif est très limité, et le simple apprentissage d'une langue articulée lui devient proprement impossible s'il n'a pas été plongé dans une culture humaine dès sa première enfance. En d'autres termes, les autres sont indispensables à notre développement cognitif et ce que nous appelons "intelligence" est autant dû aux bases génétiques qui définissent notre structure neuronale générale qu'aux interactions que nous pouvons avoir avec le monde qui nous entoure et, en particulier, avec la société humaine » (Ferber, 1995).

L'IA Distribuée ajoute donc la dimension sociale à l'IA classique.

Les capacités intellectuelles d'un être humain proviennent :

- De ses prédispositions génétiques ;
- Des interactions avec ses semblables (accointances) ;
- Des interactions avec son environnement.

Les capacités d'une machine "intelligente" devraient donc provenir :

- de ses possibilités d'inférence ;
- des interactions avec les autres machines ;
- des interactions avec son environnement.

L'IAD est née de la difficulté d'intégrer dans une même base de connaissances, l'expertise, les compétences et la connaissance de différentes entités qui communiquent et collaborent pour réaliser un but commun (Erceau & Ferber, 1991). L'IAD consiste à distribuer l'expertise au sein d'une société d'entités, appelées *agents* dont le contrôle et les données sont distribués (Guessoum, 1996). Ces agents, qui sont relativement indépendants et autonomes interagissent dans des modes simples ou complexes de coopération pour accomplir un objectif global, notamment la résolution de problèmes complexes. N.Skarmas définit l'IAD, dans «Agents as Objects with Knowledge Base State» (Skarmas, 1998), comme étant un domaine concerné par les systèmes ouverts et distribués dont les entités présentent une sorte d'intelligence et qui essaient d'accomplir des buts qui peuvent être implicites ou explicites. L'IAD a donné naissance à deux principaux domaines :

La ***résolution des problèmes distribués*** (DPS), qui s'intéresse à la décomposition d'un problème complexe en sous-problèmes et à sa résolution par des entités distribuées logiques et physiques.

Les ***systèmes multi-agents*** (SMA) qui sont concernés par la coordination du comportement des agents, qui peuvent être vus comme des entités intelligentes et autonomes (Skarmas, 1998).

SMA «versus» DPS

DPS (résolution distribuée de problème) proche de l'IA classique

- Répartir à la conception du travail nécessaire à la résolution d'un problème parmi un ensemble d'agents.
- Orienté par le problème à traiter.
- Approche descendante, centrée Agent.
⇒ contrôle centralisé, statique, modulaire

SMA (système multi-agent)

- Ensemble d'agents autonomes en interaction
- Coordonner le comportement de cet ensemble d'agents (pré-existants) pour résoudre collectivement un problème
- Dans un environnement complexe et évolutif
- Approche ascendante, centrée sociale
⇒ résolution « émergente » : à l'**exécution**

(Labidi & Lejouad 1993) et (Oliveira, 1998) affirment que l'IAD était supposée apporter une solution à des problèmes spécifiques tels que :

- ✓ la modélisation et distribution de la connaissance parmi plusieurs agents ;
- ✓ la génération de plans d'actions où la présence d'autres agents doit être considérée ;
- ✓ la résolution de conflits entre agents et la maintenance de la cohérence des décisions et plans d'actions ;
- ✓ la résolution des problèmes de communication pour permettre les interactions entre agents ;
- ✓ la résolution des problèmes spécifiques concernant l'organisation des SMAs

On voit que les systèmes multi-agents se positionnent au carrefour de la programmation (ce sont des logiciels), de l'intelligence artificielle (leur autonomie de décision), et des systèmes répartis (leur décentralisation). Historiquement, on peut replacer le concept d'agent et de système multi-agent dans l'histoire de l'intelligence artificielle et de manière duale dans l'histoire de la programmation.

5.1.4- Le monde est ouvert

La programmation classique est fermée

- Dans l'espace : le programmeur a une connaissance globale du logiciel à construire. Le principe même de l'analyse descendante d'application ou de la spécification d'application est de partir du haut (où on voit tout) puis de décomposer en parties à programmer.
- Dans le temps : bien que la notion de cycle de vie d'un logiciel introduise une dose de dynamique, il s'agit plutôt de corriger et de maintenir un logiciel spécifié une fois pour toutes.
- Dans la modalité : il existe une volonté de développer les applications de manière la plus homogène possible : mêmes personnes, mêmes logiciels de développement.
- Dans la sémantique : Les applications ont une sémantique globale et statique.
- Dans la complexité : Les applications sont bien délimitées et conçues de manière analytique. Ceci facilite le découpage en parties pour leur mise au point et l'étude de leur comportement qui est considéré comme devant être totalement prédictible.

A l'opposé, la nouvelle programmation est ouverte

L'ouverture est une propriété inhérente des systèmes d'information actuels. Elle n'est voulue par personne mais c'est un état des choses que l'on ne peut plus se permettre d'ignorer. Plutôt que d'essayer de l'enrayer, il vaut mieux essayer de la maîtriser voire en tirer avantage. Cette ouverture s'exprime :

- Dans l'espace : les systèmes d'information actuels sont intrinsèquement distribués (vision locale obligatoire) et se développent de manière agrégative.
- Dans la modalité : les deux points précédents ont pour conséquence directe que les systèmes d'information actuels sont de plus en plus hétérogènes : environnements de programmation ; points de vues adoptés sur un même sujet : fonctionnel, matériel, structurel etc. ; applications hybrides.
- Dans la sémantique : les multiples points de vue engendrent autant de mondes sémantiques hétérogènes les uns aux autres.
- Dans la complexité : les systèmes actuels présentent des interactions très intriquées ce qui les rend difficiles à prédire et à analyser

5.1.5- Domaines d'intérêts

- Intelligence Artificielle Distribuée (IAD)
- Bases de Données Distribuées (BDD)
- Systèmes d'Information Coopératifs (SIC)
- Génie logiciel
- Aide à la décision
- Apprentissage automatique

5.2- Concepts de base

5.2.1- Agent

5.2.1.1- Définitions

Ces dix dernières années, le concept d'agent a été utilisé et étudié dans plusieurs domaines. Toutefois, il n'y a encore aucun consensus, entre les différents chercheurs, quant à la définition même du mot « agent ». Selon H.S.Nwana (Nwana, 1996) et (Nwana & Ndumu,

2000) et rapporté par M.Raza (Raza, 2009), il y a au moins deux raisons qui permettent d'expliquer cette difficulté.

-La première réside dans le fait que les chercheurs, dans le domaine des agents, ne sont pas à l'origine de ce terme comme l'on été, par exemple, les chercheurs dans le domaine de la logique floue. En effet, le terme agent a été et continue d'être utilisé dans la vie de tous les jours par des personnes travaillant dans des domaines très différents.

Par exemple, on parle d'agent de voyage, d'agent immobilier, d'agent d'assurance, etc.

-La deuxième raison est que même dans la communauté des chercheurs sur les agents logiciels, le mot « agent » est utilisé pour décrire des systèmes très différents les uns des autres. Pour ajouter à la confusion, les chercheurs sont allés même jusqu'à inventer plusieurs synonymes au mot « agent ». Ils ont ainsi inventé, par exemple, « knowbots » (robots à base de connaissances), « softbots » (robots logiciel), taskbots (robots à base de tâche), « userbots » (robots pour utilisateur), robots, agents personnels, agents autonomes, assistants personnels, etc. Il est vrai qu'une telle prolifération de termes trouve sa justification dans le fait que les agents peuvent prendre différentes formes physiques (robot ou agent logiciel) et qu'ils peuvent aussi jouer plusieurs rôles.

Cela dit, il est tout de même important de s'entendre sur une définition du terme agent pour que les exposés qui suivent dans ce mémoire aient un sens. La définition que nous avons adoptée, et qui semble couvrir les caractéristiques des agents que nous avons développés, est celle proposée par Jennings, Sycara et Wooldridge (Jennings & all, 1998) :

A ce propos, Carl Hewitt fit remarquer (lors du troisième international workshop sur l'IAD), que la question qu'est ce qu'un agent ? est aussi embarrassante pour la communauté informatique que la question qu'est ce que l'intelligence ? est embarrassante pour la communauté d'intelligence artificielle.

Etant donné les origines diverses du concept agent, nous ne pouvons pas donner une seule définition au terme agent. En effet, plusieurs définitions ont été proposées par différents auteurs pour clarifier ce concept.

Déf1 : « les agents peuvent être vus comme des unités intelligentes et autonomes » (Skarmas, 1998)

Déf2 : « les objets qui pensent » (Magendaz, 1995)

Déf3 : Un agent est un système informatique, situé dans un environnement, qui agit d'une façon autonome et flexible pour atteindre les objectifs pour lesquels il a été conçu.

Déf4 : Un agent est une entité logicielle ou physique à qui est attribuée une certaine mission qu'elle est capable d'accomplir de manière autonome et en coopération avec d'autres agents.

Déf5 : Un agent intelligent tout ce qui perçoit son environnement à l'aide de ses capteurs et qui agit sur son environnement à l'aide de ses effecteurs. (Chaib-Draa & all, 2001)

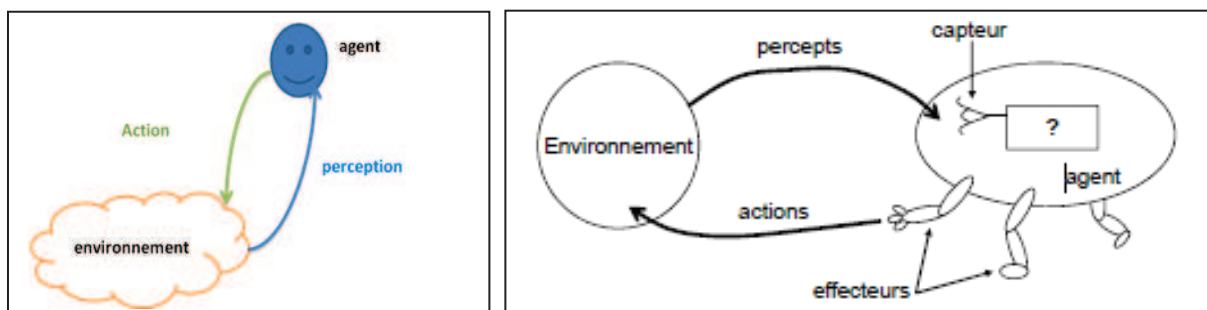


Figure 5.4 : L'environnement d'un agent

Déf6 : Le terme agent représente une entité intelligente qui agit de manière rationnelle et intentionnelle en fonction de ses buts et de l'état courant de ses connaissances (Demazeau & Müller, 1991).

Déf7 : un agent est entité dotée d'un état mental, qui représente ses connaissances, croyances, intentions et engagements vis-à-vis de lui-même et des autres agents. (Shoham, 1993), (Cohen & Levesque, 1995), (Wooldridge & Jennings, 1994), (Wooldridge, 1999) (Wooldridge, 2002).

Déf8 : Selon Jacques (Ferber, 1995) Qu'est ce qu'un agent? Comme dans tous les domaines porteurs, le terme agent est utilisé de manière assez vague. Cependant on peut dégager une définition minimale commune qui est approximativement la suivante:

: On appelle agent une entité physique ou virtuelle

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

Chacun des termes de cette définition est important

Une entité physique est quelque chose qui agit dans le monde réel: un robot, un avion ou une voiture sont des exemples d'entités physiques.

En revanche, un composant logiciel, un module informatique sont des entités virtuelles, car elles n'existent pas physiquement.

Les agents sont capables d'agir, et non pas seulement de raisonner comme dans les systèmes d'IA classique. L'action, qui est un concept fondamental pour les systèmes multi-agents, repose sur le fait que les agents accomplissent des actions qui vont modifier l'environnement des agents et donc leurs prises de décision futures.

Ils peuvent aussi communiquer entre eux, et c'est d'ailleurs là l'un des modes principaux d'interaction existant entre les agents.

Ils agissent dans un environnement

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

besoin d'éléments qui lui sont extérieurs pour survivre) et un système fermé (car les échanges qu'il a avec l'extérieur sont très étroitement réglementés).

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

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

5.2.1.2- Des Objets aux Agents

D'un point de vue informatique, l'approche multi-agent peut être considérée comme une évolution du paradigme orienté-objet. Du point de vue conceptuel, un objet est simplement une structure de données à laquelle sont associées des fonctions. Les agents sont des entités autonomes, ce qui signifie que leur comportement ne dépend pas d'une pression extérieure, contrairement aux objets.

-Agent: entité autonome interagissant avec son environnement

-Objet: entité passive possédant un état et sur lequel on peut effectuer des opérations.

(Chaib-draa, 2010)

-Un agent est à un degré d'abstraction plus élevé qu'un objet.

Un agent peut être constitué de plusieurs objets.

-C'est un paradigme de programmation mettant en évidence l'autonomie et les interactions.

(Programmation orientée-agent)

Similarités :

Possèdent un «état interne»,

Des unités de comportement modulaires (méthodes/compétences),

Communiquent par envoi de messages,

Peuvent agir pour modifier leur état,

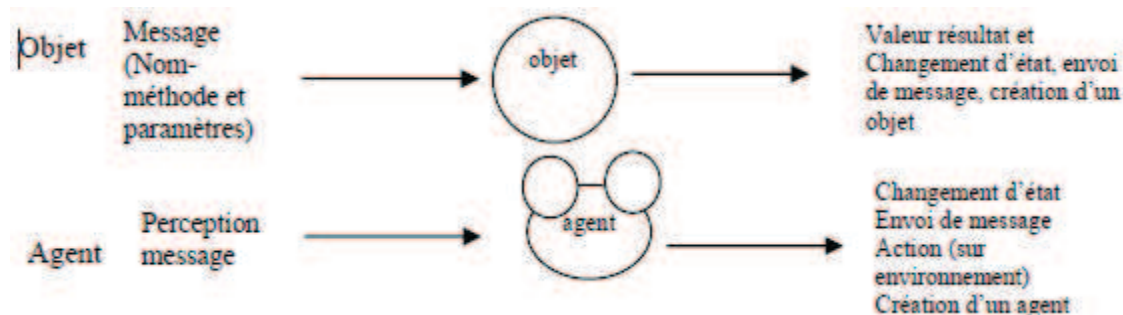


Figure 5.5 : Objet «versus» Agent

Différences entre objets et agents (Briot & Demazeau, 2001-2002)

Objet	Agent
Pas d'autonomie: l'objet qui reçoit un appel de méthode exécute celle-ci (pas de proactivité, de réactivité)	Autonomie de contrôle: l'agent décide de son comportement en fonction de son état, croyances, connaissances, perceptions de l'environnement, requêtes des autres

Peu de socialité: interaction rigide (pas d'évolution dans le temps) et simple	Socialité: composante très importante, complexité des interactions, des organisations
	Notion d'environnement importante et plus complexe

Tableau 5.1 : Différences entre objets et agents

5.2.2- Système Multi-Agents

5.2.2.1- Qu'est-ce qu'un système multi-agents ?

Précédemment on a présenté des systèmes où il n'y avait qu'un seul agent mais, dans la plupart des situations réelles, l'agent n'est pas seul dans son environnement, il y a d'autres agents présents autour de lui. Il nous faut donc aborder des systèmes où plusieurs agents doivent interagir entre eux pour effectuer leurs tâches. De tels systèmes sont appelés « systèmes multi-agents » et ils possèdent les caractéristiques principales (Jennings & all, 1998) suivantes :

- chaque agent a des informations ou des capacités de résolution de problèmes incomplètes, donc chaque agent a un point de vue limité ;
- il n'y a pas de contrôle global du système ; les données sont décentralisées ;
- les calculs sont asynchrones.

La définition d'un système multi-agents (avec son acronyme SMA, et MAS pour « multi-agent system » en anglais) est plus immédiate : «Un ensemble d'agents qui agissent (et interagissent) dans un environnement commun ». Nous ne faisons que suivre ici la définition usuelle du terme système : « un ensemble organisé d'éléments ». Cela signifie que dans un système multi-agent, il existe une ou plusieurs organisations qui structurent les règles de cohabitation et de travail collectif entre agents (définition des différents rôles, partages de ressources, dépendances entre tâches, protocoles de coordination, de résolution de conflits, etc.). Dans un même système, il existe en général plusieurs organisations et un même agent peut appartenir à plusieurs simultanément. Des exemples d'organisations d'agents dans le monde réel sont une organisation économique telle qu'une entreprise, mais aussi une organisation animale telle qu'une fourmilière.

Selon (Ferber, 1995) : On appelle système multi-agent (ou SMA), un système composé des éléments suivants:

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

5.2.2.2- Utilité des systèmes multi-agents

Certains domaines requièrent l'utilisation de plusieurs entités, par conséquent, les systèmes multi-agents sont très bien adaptés à ce type de situations. Par exemple, il y a des systèmes qui sont géographiquement distribués comme la coordination entre plusieurs frégates, le contrôle

aérien, les bases de données coopératives distribuées, etc. Tous ces domaines sont par définition distribués, par conséquent, les systèmes multiagents procurent une façon facile et efficace de les modéliser.

Une autre situation, où les systèmes multiagents sont requis, est lorsque les différents systèmes et les données qui s'y rattachent appartiennent à des organisations indépendantes qui veulent garder leurs informations privées et sécurisées pour des raisons concurrentielles. Par exemple, la majorité des missions maritimes se font maintenant en collaboration avec plusieurs pays. Donc, il y a plusieurs bateaux de pays différents qui doivent agir ensemble. Pour concevoir un système qui permettrait aux bateaux de se coordonner, il faudrait avoir des informations sur toutes les caractéristiques de chacun des bateaux. Par contre, aucun pays ne voudra donner ces informations puisqu'elles sont considérées comme des secrets militaires. De plus, les façons de faire diffèrent d'un pays à l'autre. Une solution à ce problème est de permettre à chaque pays de concevoir ses propres agents qui représenteront adéquatement ses buts et ses intérêts. Par la suite, ces agents pourront communiquer ensemble pour coordonner l'ensemble de la mission. Dans ce cas, les agents ne se transmettront que les informations nécessaires à une bonne coordination des bateaux.

Même si le domaine ne requiert pas l'utilisation des systèmes multiagents, il existe tout de même de bons avantages à utiliser ce type de système. Ainsi, ils peuvent s'avérer bien utiles pour des problèmes possédant de multiples méthodes de résolution, de multiples perspectives et/ou de multiples solveurs. En particulier, les systèmes multiagents sont très utiles pour modéliser le raisonnement humain à l'intérieur de grandes simulations de combats aériens.

Ils possèdent également les avantages traditionnels de la résolution distribuée et concurrente de problèmes:

- La modularité permet de rendre la programmation plus simple. Elle permet, de plus, aux systèmes multiagents d'être facilement extensibles, parce qu'il est plus facile d'ajouter de nouveaux agents à un système multiagents que d'ajouter de nouvelles capacités à un système monolithique.

- La vitesse est principalement due au parallélisme, car plusieurs agents peuvent travailler en même temps pour la résolution d'un problème.

La fiabilité peut être également atteinte, dans la mesure où le contrôle et les responsabilités étant partagés entre les différents agents, le système peut tolérer la défaillance d'un ou de plusieurs agents. Si une seule entité contrôle tout, alors une seule défaillance de cette entité fera en sorte que tout le système tombera en panne.

Finalement, les systèmes multiagents héritent aussi des bénéfices envisageables du domaine de l'intelligence artificielle comme le traitement symbolique (au niveau des connaissances), la facilité de maintenance, la réutilisation et la portabilité.

5.2.2.3- Un premier exemple

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

Plutôt que d'envoyer un seul robot multi-spécialisé (locomotion, vision, prélèvement d'échantillons, analyse, communication avec la Terre, etc.), il peut être intéressant d'y substituer plusieurs robots. Ils pourront être spécialisés (par exemple, un robot spécialisé dans les prélèvements, un autre dans les analyses - ce dernier étant éventuellement immobile -, etc.). Mais ils devront coopérer et coordonner leurs activités à partir de leurs connaissances et comportements locaux. Une telle spécialisation selon les expertises rend une conception et

une construction plus aisées qu'un robot unique qui doit savoir tout faire. Une telle organisation offre également une meilleure efficacité potentielle (du fait des tâches pouvant être effectuées en parallèle). De plus si l'on assure une certaine redondance (par exemple plusieurs robots d'analyse), cela assure une certaine robustesse à l'ensemble du système en cas de panne d'un de ses éléments. Ceci est donc un premier exemple de système multi-agent. Dans ce cas, les agents sont robotiques (on parle de robotique collective). Un autre exemple est celui de la surveillance de réseaux où l'on délègue des tâches de surveillance, détection d'anomalies, diagnostic et réparations à différents agents logiciels coopératifs.

5.2.2.4- Vue intuitive d'un Agent dans un SMA

Un SMA peut-être :

Ouvert : les agents y entrent et en sortent librement (ex: un café)

Fermé : l'ensemble d'agents reste le même (ex: un match de football)

Homogène : tous les agents sont construit sur le même modèle (ex: une réunion de travail, une colonie de fourmis)

Hétérogène : des agents de modèles différents, de granularité différentes (ex: l'organisation hospitalière)

5.2.2.5- Variables globales et locales et les SMA

En programmation classique les variables globales sont accessibles à tous et les variables locales ne sont accessibles à personne.

Global X, Y

FuncfooLocal a,b

... a ...

... X ...

Funcbar Local u,v,w

... u,v...

... X ...

Dans un SMA rien n'est complètement global

L'environnement étant vaste (sans compter le problème de la représentation récursive —je sais que je sais que je sais... —) et ouvert, il n'est pas possible en un lieu donné (par exemple, dans un agent) de stocker toute la représentation du monde. Par contre, un agent peut se déplacer ou encore interagir avec d'autres agents qui sont dans son voisinage pour explorer l'environnement.

Dans un SMA rien n'est complètement local

Pour un agent donné, toutes ses entités (informations, processus, buts, ...) sont locales mais elles restent accessibles à l'introspection par d'autres agents. Le moyen d'accéder à cette information passe par les interactions entre les agents.

5.2.2.6- Niveaux d'organisation

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

a- Le niveau micro-social, où l'on s'intéresse essentiellement aux interactions entre agents et aux différentes formes de liaison qui existent entre deux ou un petit nombre d'agents. C'est à ce niveau que la plupart des études ont été généralement entreprises en intelligence artificielle distribuée.

b- Le niveau des groupes, où l'on s'intéresse aux structures intermédiaires qui interviennent dans la composition d'une organisation plus complète. A ce niveau, on étudie les différenciations des rôles et des activités des agents, l'émergence de structures organisatrices entre agents et le problème général de l'agrégation des agents lors de la constitution d'organisations.

c- Le niveau des sociétés globales (ou populations) où l'intérêt se porte surtout sur la dynamique d'un grand nombre d'agents, ainsi que sur la structure générale du système et son évolution. Les recherches se situant dans le cadre de la vie artificielle se situent assez souvent à ce niveau.

5.2.3- Propriétés d'un agent intelligent

Un agent peut être caractérisé par plusieurs propriétés, nous pouvons citer parmi elles :

5.2.3.1- Autonomie

Wooldridge et Jennings (Wooldridge & Jennings, 1994) définissent l'autonomie comme étant la capacité pour un agent d'opérer de manière autonome sans une intervention directe d'humains ou d'autres agents et de contrôler ses actions et son état interne. Quant à C.CastelFranchi dans (Castelfranchi, 1995), définit un agent autonome comme étant un agent qui a ses propres buts et qui est capable de décider des buts à poursuivre, comment les atteindre et résoudre les conflits internes relatives aux buts choisis. Pour P.R.Cohen, H.J.Levesque dans (Cohen & Levesque, 1995) et Y.Shoham dans (Shoham 1993), un agent est capable :

- d'agir selon ses intentions ;
- d'adopter les buts qu'il croit réalisables et d'abandonner ceux qu'il croit irréalisables ;
- de planifier ses propres actions et de tenir compte de celles des autres ;
- de raisonner non seulement sur ses connaissances mais aussi sur celles des autres.

5.2.3.2- Réactivité

Les agents perçoivent leur environnement et réagissent aux changements qui s'y produisent (Wooldridge & Jennings, 1994).

La réactivité signifie aussi la capacité qu'a un agent de modifier son comportement lorsque les conditions environnementales changent (Oliveira, 1998).

Un objet est une entité passive(ou réactive), si personne ne demande la valeur d'un attribut ou n'active une méthode de l'objet, il ne se passe rien (Sansonnnet, 2002).

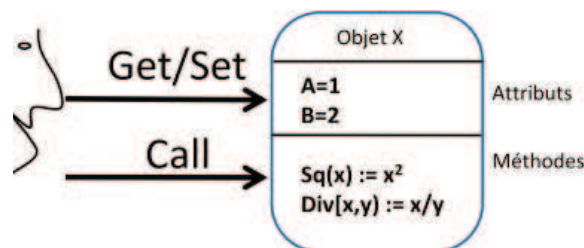


Figure 5.6 : Réactivité

5.2.3.3- Proactivité

Les agents n'agissent pas simplement en réponse à leur environnement, ils sont capables d'exhiber un comportement guidé par des buts en prenant des initiatives (Wooldridge & Jennings, 1994). La proactivité est la capacité d'un agent d'anticiper des situations et de changer son cours d'action pour les éviter (Oliveira, 1998).

Un agent possède, en plus des attributs et méthodes, des processus internes qui fonctionnent même en l'absence de sollicitations externes. Un agent peut donc agir même si personne ne lui demande rien.

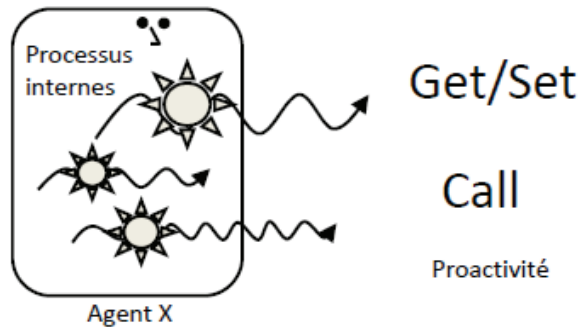


Figure 5.7 : Proactivité

5.2.3.4- Adaptabilité

Face à un environnement perpétuellement changeant, un agent doit constamment modifier le plan qu'il poursuit pour atteindre un but. Pour cela :

Il doit, de manière continue, percevoir et évaluer la situation (contexte) de son action.

Construire des représentations en cours même de fonctionnement (c'est à dire être capable d'apprendre).

Élaborer des plans dynamiques qui lance des processus internes ou au contraire les stoppent.

L'adaptabilité est la capacité d'un agent de s'adapter à l'environnement dans lequel il est situé (Oliveira, 1998). Un agent adaptatif est un agent capable de contrôler ses aptitudes (communicationnelles et comportementales) selon l'environnement dans lequel il évolue et selon l'agent avec lequel il interagit (Guessoum, 1996).

L'adaptabilité est une propriété très importante pour un agent qui évolue dans un environnement dynamique.

5.2.3.5- Sociabilité

La sociabilité est la capacité d'un agent de s'intégrer dans un environnement peuplé d'agents avec qui il échange des messages pour accomplir un but (Oliveira, 1998).

Nous reviendrons sur cette propriété lorsque nous aborderons les propriétés d'un SMA car elle intègre d'autres propriétés telles que la communication, la coopération et la délégation.

5.2.3.6- Apprentissage

L'apprentissage est une propriété assez particulière car un agent n'est pas forcément une entité capable d'apprendre. Les agents peuvent avoir à apprendre lorsqu'ils réagissent et/ou interagissent avec leur environnement externe (Nwana, 1996).

L'apprentissage est une propriété qui fournit aux systèmes la capacité d'acquérir la compréhension de certains comportements au cours du temps, sans nécessiter que ces comportements soient programmés manuellement (Guessoum & all, 2006).

Un agent a la faculté d'apprendre s'il est capable d'utiliser de nouvelles connaissances pour modifier son comportement (Cheikhrouhou & all, 1998).

Bien que l'apprentissage soit un attribut clé de l'intelligence (Nwana, 1996), il existe très peu d'agents qui ont cette capacité.

5.2.3.7- Sécurité

La sécurité est une propriété importante, notamment dans le contexte de ce travail, car elle permet de garantir que lorsque l'on interagit avec un agent, que cet agent n'a pas été corrompu par un virus, par de fausses croyances ou par des connaissances qui n'ont pas de sens (Oliveira, 1998).

5.2.4- Propriétés des systèmes multi-agents

5.2.4.1- Interactions entre agents

Jacques Ferber donne la définition suivante de l'interaction : " Une interaction est la mise en relation dynamique de deux ou plusieurs agents par le biais d'un ensemble d'actions réciproques. Les interactions sont non seulement la conséquence d'actions effectuées par plusieurs agents en même temps, mais aussi l'élément nécessaire à la constitution d'organisations sociales. " (Ferber, 1995).

Les systèmes multiagents ont surtout l'avantage de faire intervenir des schémas d'interaction sophistiqués. Ils peuvent ainsi coexister, être en compétition ou coopérer.

S'ils ne font que coexister, alors chaque agent ne considère les autres agents que comme des composantes de l'environnement, au même titre que toutes les autres composantes.

Si les agents ont une représentation physique, les autres agents ne seront vus que comme des obstacles que l'agent doit éviter. Il s'ensuit qu'il n'y a aucune communication directe entre les agents. En fait, il peut y avoir une certaine forme de communication indirecte parce que les agents peuvent se percevoir les uns les autres. Le but visé n'est toutefois pas de communiquer avec l'autre. Ces informations ne servent qu'à mieux éviter les autres agents. Par exemple, si l'on considère une personne marchant dans une foule d'étrangers, elle communique avec les autres à l'aide de gestes ou de mouvements, mais uniquement dans le but de pouvoir circuler sans accrocher tout le monde. (Chaib-Draa & Gageut, 2002)

S'ils sont en compétition, alors le but de chaque agent est de maximiser sa propre satisfaction, ce qui se fait généralement aux dépens des autres agents. La situation de compétition la plus fréquente se produit lorsque plusieurs agents veulent utiliser ou acquérir la même ressource. Les agents doivent donc pouvoir communiquer entre eux pour résoudre le conflit. Cette communication prend habituellement la forme d'une négociation. Les agents se transmettent des propositions et des contre-propositions jusqu'à ce qu'ils arrivent à une entente ou qu'ils se rendent compte qu'une entente est impossible. Ce type de communication demande un protocole de négociation et un langage de communication de haut niveau du type de KQML (Finin & Fritzson, 1994) ou FIPA-ACL (Fipa, 2000) pour permettre une certaine structure dans la négociation.

S'ils sont en coopération, alors le but des agents n'est plus seulement de maximiser sa propre satisfaction mais aussi de contribuer à la réussite du groupe. Les agents travaillent ensemble à la résolution d'un problème commun. Dans ce type de système, les agents communiquent ensemble, à l'aide de messages plus ou moins sophistiqués, dans le but d'améliorer la performance du groupe. Ils peuvent s'échanger des informations sur l'environnement pour augmenter leurs perceptions individuelles, ou bien se transmettre leurs intentions pour que les agents puissent avoir une idée de ce que les autres font. Somme toute, les communications servent aux agents à améliorer leur coordination, c'est-à-dire à organiser la résolution du

problème de telle sorte que les interactions nuisibles soient évitées et/ou que les interactions bénéfiques soient exploitées.

Nous avons abordé la *sociabilité* dans la section précédente mais en réalité cette propriété est centrée sur les interactions entre agents (Guessoum, 1996). Nous allons donc voir maintenant les différentes propriétés qui sont impliquées dans la sociabilité.

5.2.4.2- Coopération

Pour Ferber (Ferber, 1995), pour que plusieurs agents soient dans une situation de coopération, il faut que l'une des deux conditions suivantes soit vérifiée :

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

La coopération entre agents peut être (Oliveira, 1998) :

Une approche multiagents pour la gestion de sécurité

- soit **implicite** et dans ce cas les agents ont un but commun implicite qu'ils doivent atteindre en exécutant des actions indépendantes. Dans ce type de coopération, la communication entre les agents est facultative.
- soit **explicite** et dans ce cas, les agents exécutent des actions qui leur permettent d'achever non seulement leurs propres buts mais aussi les buts des autres agents.

Ce type de coopération nécessite une communication entre les agents pour prendre connaissance des buts des autres agents.

5.2.4.3- Coordination

Dans un système multi-agents, la coordination des actions des différents agents permet d'assurer une cohérence du système. Il existe plusieurs mécanismes de coordinations parmi lesquels, nous retrouvons : l'organisation, la planification et la synchronisation (Guessoum, 1996), (Oliveira, 1998). Nous définissons ces trois mécanismes.

a- L'organisation :

Une organisation représente un groupe d'agents qui travaillent ensemble afin de réaliser une ou plusieurs tâches (Guessoum, 1996). De nombreux travaux sur l'organisation ont été proposés. T.Bouron (Bourron, 1992) les a regroupés en deux classes où l'organisation est définie :

- Soit comme une structure externe aux agents et elle est représentée par un objet ou un agent.
- Soit comme un objet abstrait dont la représentation est distribuée parmi les membres de l'organisation.

Parmi les travaux existants, nous pouvons citer le *réseau contractuel* introduit par R.Smith (Smith, 1980). Le *réseau contractuel* est un mécanisme d'allocation de tâches fondé sur la notion d'appel d'offre (Ferber, 1995).

Dans ce modèle, un agent peut avoir deux rôles par rapport à une tâche (Guessoum, 1996) : *manager* ou *contractant*. L'appel d'offre s'effectue en quatre étapes :

- **appel d'offre** : le *manager* décompose une tâche en sous-tâches. Il recherche ensuite des *contractants* pour les réaliser en faisant une annonce de tâches à tous les agents du système ;
- **envoi de propositions** : les agents élaborent une proposition et font une offre au *manager* ;
- **attribution du marché** : le *manager* évalue les propositions, attribue la tâche à un ou plusieurs agents et informe les autres agents de son choix ;

-
- **établissement du contrat** : l'agent ou les agents sélectionnés deviennent *contractants* et informent le *manager* qu'ils s'engagent à réaliser la tâche.

b- La planification :

La planification multiagents est une autre approche de coordination dans les systèmes à base d'agents. Afin d'éviter des actions conflictuelles ou inconsistantes, les agents construisent un plan multi-agents qui détaille toutes les actions futures et les interactions nécessaires pour atteindre leurs buts (Oliveira, 1998).

La planification dans les SMAs, se décompose en trois étapes (Ferber, 1995) :

- la construction de plans,
- la synchronisation et coordination des plans,
- l'exécution de ces plans.

Il existe deux types de planification, dans les SMAs :

- la **planification centralisée** pour agents multiples qui suppose l'existence d'une vue globale du plan (Guessoum, 1996). Dans cette approche, il existe un seul agent capable de planifier et d'organiser les actions pour l'ensemble des agents (Ferber, 1995).
- la **planification distribuée** où chaque agent planifie individuellement ses actions en fonction de ses propres buts (Ferber, 1995). Les différents agents se communiquent ensuite leurs plans partiels afin de détecter et d'éviter les conflits éventuels.

c- La synchronisation :

La synchronisation est le "bas niveau" de la coordination où sont implémentés les mécanismes de base permettant aux différentes actions de s'articuler correctement. Elle permet de synchroniser l'enchaînement des actions des différents agents (Ferber, 1995).

J. Ferber répertorie deux types de synchronisation :

- La **synchronisation par mouvement** utilisée lorsque plusieurs éléments doivent se déplacer ensemble. Dans ce type de synchronisation, il s'agit de coordonner le rythme et le positionnement dans le temps d'actions en fonction des événements qui se produisent ;
- La **synchronisation d'accès à une ressource** utilisée lorsque plusieurs agents doivent partager une ressource.

5.2.4.4- La compétition

La compétition entre agents peut avoir plusieurs sources. Les buts des agents peuvent être incompatibles : dans une situation de jeu, chacun cherche à être gagnant mais la réalisation de ce but pour un des joueurs rend impossible la réalisation du but des autres joueurs. Les ressources dont les agents ont besoin peuvent être rares et l'utilisation d'une ressource par un des agents peut empêcher un autre agent de réaliser son but. Par exemple, si l'imprimante est utilisée je dois attendre pour imprimer mon article et je ne pourrai pas le poster avant le départ du courrier. La compétition crée des situations de conflit qui peuvent être résolues par la lutte ou par la coordination (Drogoul, 1993).

5.2.4.5- Délégation

La délégation est la capacité d'un agent à exécuter des tâches pour le compte d'un tiers. Elle permet ainsi à un agent, qui ne peut pas atteindre ses buts par manque de ressources, de compétences ou dans le but d'alléger sa tâche, de demander à d'autres agents d'achever des buts pour son compte. Cette propriété est cruciale lorsque la coordination est supportée par une structure organisationnelle (Oliveira, 1998) et que l'environnement du SMA varie et

nécessite par conséquent une redistribution et mise à jour des tâches que doivent accomplir les différents agents (ce qui se traduit par une délégation de nouveaux buts). Dans une organisation hiérarchique par exemple, lorsque le SMA doit atteindre de nouveaux objectifs, elle permet à une entité gestionnaire de déléguer des sous-tâches à des agents sous-jacents et de les modifier.

La délégation est beaucoup utilisée dans la gestion de réseau où elle permet une flexibilité du système qui l'utilise (Magendaz, 1995).

5.2.4.6- Communication

Dans un SMA, les agents communiquent entre eux en s'échangeant des informations via un langage de communication agent (Wooldridge & Jennings, 1994). Le langage, le plus utilisé aujourd'hui est KQML (Knowledge Query and Manipulation Language) qui est un langage de haut niveau utilisant une liste de types de messages, appelés performatifs (Ferber, 1995). Les agents peuvent également communiquer en utilisant d'autres mécanismes tels que le mécanisme de tableau noir (Guessoum 1996).

5.2.4.7- Une Recherche de Compromis

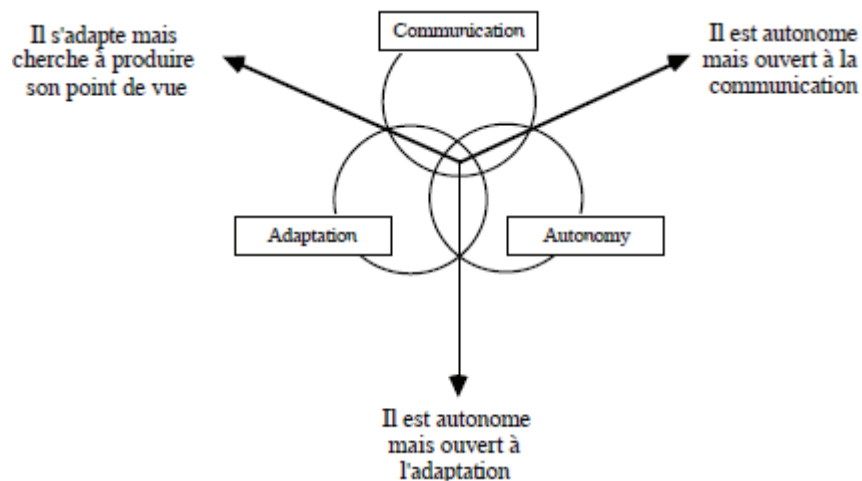


Figure 5.8 : Le compromis recherché (SMA)

5.3- Les différents modèles d'agents (Architecture)

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

Deux grandes écoles émergent, celles des *Agents cognitifs* et celles des *réactifs* :

- les systèmes *délibératifs*, dits aussi *cognitifs*,
- les systèmes *réactifs*. (Boissier, 2001)

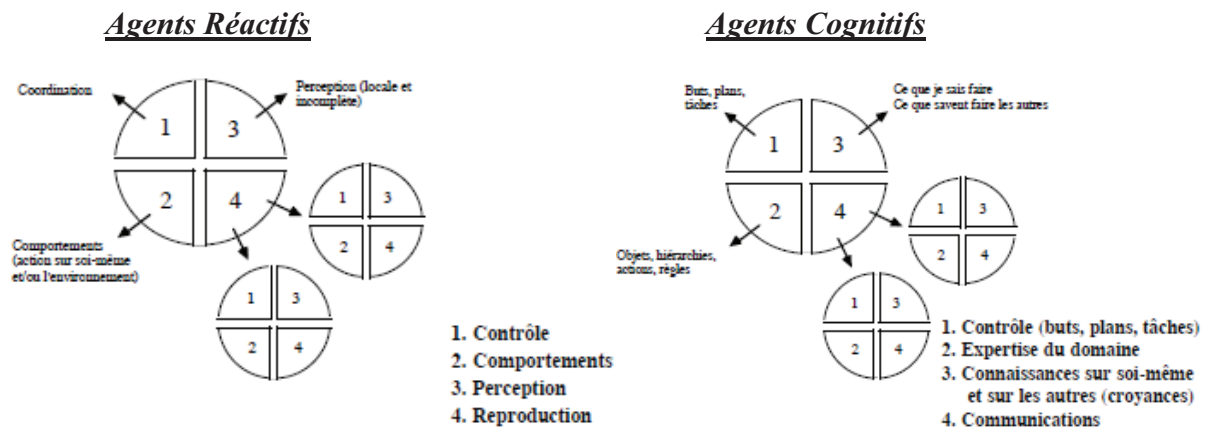


Figure 5.9 : Agents réactifs et cognitifs

Ces deux conceptions ont donné lieu à deux écoles de pensée. La première, l'école "cognitive", est la plus représentée dans le domaine que l'on appelle intelligence artificielle distribuée (IAD) car elle trouve son origine dans la volonté de faire communiquer et coopérer des systèmes experts classiques. Dans ce cadre, un système multi-agent est composé d'un petit nombre d'agents "intelligent". Chaque agent dispose d'une base de connaissance comprenant l'ensemble des informations et des savoir-faire nécessaires à la réalisation de sa tâche et à la gestion des interactions avec les autres agents et avec son environnement. On dit aussi que les agents sont "intentionnels", c'est-à-dire qu'ils possèdent des buts et des plans explicites leur permettant d'accomplir leurs buts. Dans ce cadre, les problèmes de coopération ressemblent étonnamment à ceux de petits groupes d'individus, qui doivent coordonner leur activité et sont parfois amenés à négocier entre eux pour résoudre leurs conflits (Bond & Gasser, 1988), (Demazeau & Müller, 1991), (Chaib-Draa & all, 1992), (Briot & all, 2006).

Les analogies sont alors sociales et nombre de chercheurs dans ce domaine s'appuient sur les travaux de sociologie et en particulier sur la sociologie des organisations et des petits groupes. L'autre tendance, l'école "réactive", prétend au contraire qu'il n'est pas nécessaire que les agents soient intelligents individuellement pour que le système ait un comportement global intelligent (Deneubourg & all, 1991), (Ferber & Drogoul, 1992). Des mécanismes de réaction aux événements, ne prenant en compte ni une explicitation des buts, ni des mécanismes de planification, peuvent alors résoudre des problèmes qualifiés de complexes. L'exemple le plus manifeste d'organisation émergente est celle de la fourmilière (Corbara & all, 1993): alors que toutes les fourmis se situent sur un plan d'égalité et qu'aucune d'entre elles ne possède de pouvoir d'autorité stricte sur les autres, les actions des fourmis se coordonnent de manière que la colonie survive et fasse donc face à des problèmes complexes tels que ceux posés par la recherche de nourriture, les soins à donner aux œufs et aux larves, la construction de nids, la reproduction, etc.

Cependant, dans la littérature, sont répertoriés quatre différents types de modèles d'agents, à savoir :

- les agents à réflexes simples,
- les agents conservant une trace du monde,
- les agents ayant des buts,
- les agents utilisant une fonction d'utilité. (Russell & Norvig, 1995)

Les deux premiers types d'agents sont considérés comme des agents réactifs et les deux derniers types sont considérés comme des agents délibératifs ou cognitifs.

Il existe un grand nombre de typologies d'agents mais celle que nous retenons, compte tenu du problème à modéliser, est fondée sur le processus de prise de décision de l'agent.

Elle distingue trois types d'agents : réactifs, cognitifs ou délibératifs et hybrides.

5.3.1- Les agents réactifs

L'idée d'architectures d'*agents réactifs* a été essentiellement introduite par Brooks (Brooks 1991). Les architectures *réactives*, dites aussi *comportementales*, se caractérisent par des agents qui ont la capacité de réagir rapidement à des problèmes simples, qui ne nécessitent pas un haut niveau de raisonnement, comme son nom l'indique, un agent réactif ne fait que réagir aux changements qui surviennent dans leur environnement ou aux messages provenant des autres agents. Autrement dit, un tel agent ne fait ni délibération ni planification, il se contente simplement d'acquiescer des perceptions et de réagir à celles-ci en appliquant certaines règles prédéfinies. Étant donné qu'il n'y a pratiquement pas de raisonnement, ces agents peuvent agir et réagir très rapidement.

Il convient de remarquer que les humains aussi utilisent cette manière d'agir. Dans plusieurs situations, il est souvent préférable de ne pas penser et de réagir immédiatement.

Par exemple, lorsqu'une personne met la main sur une plaque très chaude, elle ne commence pas à se demander si c'est chaud, si ça fait mal et s'il faut ou non qu'elle retire sa main. Dans ce cas, elle retire sa main immédiatement, sans réfléchir, et c'est cette rapidité de réaction qui lui permet de diminuer les blessures. Cet exemple montre bien que ce type de comportement réflexe est essentiel pour les êtres humains. De la même manière, il est aussi essentiel pour les agents s'ils veulent pouvoir agir dans le monde réel.

En effet, leurs décisions sont essentiellement basées sur un nombre très limité d'informations et sur des règles simples de type *situation - action* (Müller, 1996). Ce type d'architectures ne nécessite aucune représentation symbolique de l'environnement réel. Ces agents ne disposent que d'un protocole de communication et d'un langage de communication réduits (Guessoum, 1996). Dans un système à base d'agents réactifs, l'intelligence émerge de l'interaction des agents avec leur environnement.

Les deux sous-sections suivantes décrivent deux modèles qui peuvent servir à la conception d'agents réactifs.

5.3.1.1- Agents à réflexes simples

Ce type d'agent agit uniquement en se basant sur ses perceptions courantes. Il utilise un ensemble de règles prédéfinies, du type **Si condition alors action**, pour choisir ses actions. Par exemple, pour un agent en charge du contrôle de la défense d'une frégate, on pourrait avoir la règle suivante :

Si missile-en-direction-de-la-frégate alors lancer-missile-d'interception

Comme on peut le constater, ces règles permettent d'avoir un lien direct entre les perceptions de l'agent et ses actions. Le comportement de l'agent est donc très rapide, mais peu réfléchi. À chaque fois, l'agent ne fait qu'exécuter l'action correspondant à la règle activée par ses perceptions.

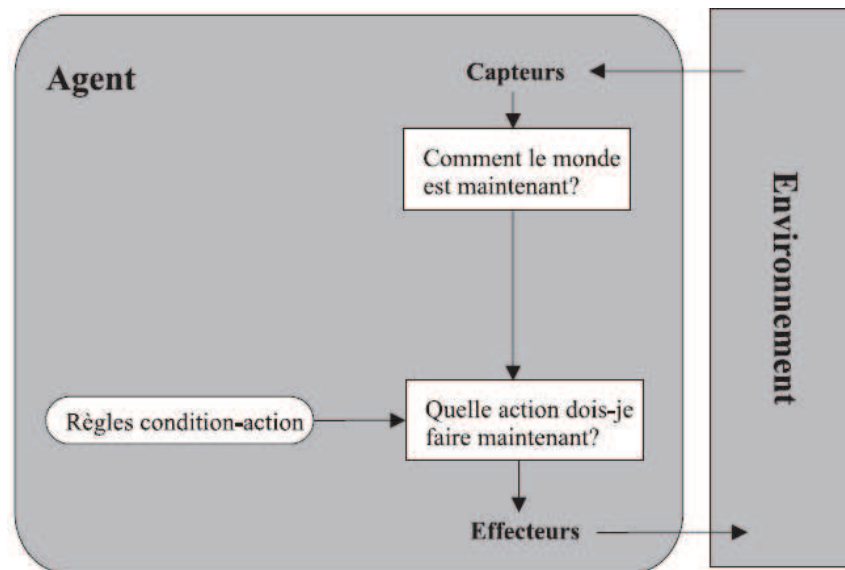


Figure 5.10 : Schéma d'un agent à réflexes simples (Russell & Norvig, 1995)

La figure montre l'architecture d'un agent à réflexes simples. Les rectangles représentent l'état interne de l'agent dans son processus de décision et les ovales représentent les informations qui sont utilisées dans le processus. L'agent se bâtit une représentation du monde à l'aide de ses perceptions lui venant de ses capteurs. Par la suite, il utilise ses règles pour choisir l'action qu'il doit effectuer selon ce qu'il perçoit de l'environnement.

5.3.1.2- Agents conservant une trace du monde

Le type d'agent qui a été décrit précédemment, ne peut fonctionner que si un tel agent peut choisir ses actions en se basant uniquement sur sa perception actuelle. Par exemple, si le radar de la frégate détecte un missile et que l'instant d'après, il le perd de vue, dû à un obstacle, cela ne signifie nullement qu'il n'y a plus de missile. Dès lors, l'agent en charge du contrôle de la frégate doit tenir compte de ce missile dans le choix de ses actions et ce, même si le radar ne détecte plus le missile.

Le problème que nous venons de mentionner survient parce que les capteurs de l'agent ne fournissent pas une vue complète du monde. Pour régler ce problème, l'agent doit maintenir des informations internes sur l'état du monde dans le but d'être capable de distinguer deux situations qui ont des perceptions identiques, mais qui, en réalité, sont différentes. L'agent doit pouvoir faire la différence entre un état où il n'y a pas de missile et un état où le missile est caché, même si ses capteurs lui fournissent exactement les mêmes informations dans les deux cas.

Pour que l'agent puisse faire évoluer ses informations internes sur l'état du monde, il a besoin de deux types d'information. Tout d'abord, il doit avoir des informations sur la manière dont le monde évolue, indépendamment de l'agent. Par exemple, il doit savoir que si un missile avance à une vitesse de 300 m/s, alors 5 secondes plus tard, il aura parcouru 1500 mètres. L'agent doit avoir ensuite des informations sur la manière dont ses propres actions affectent le monde autour de lui. Si la frégate tourne, l'agent doit savoir que tout ce qui l'entoure tourne aussi. Il doit donc mettre à jour la position relative de tous les objets autour de la frégate.

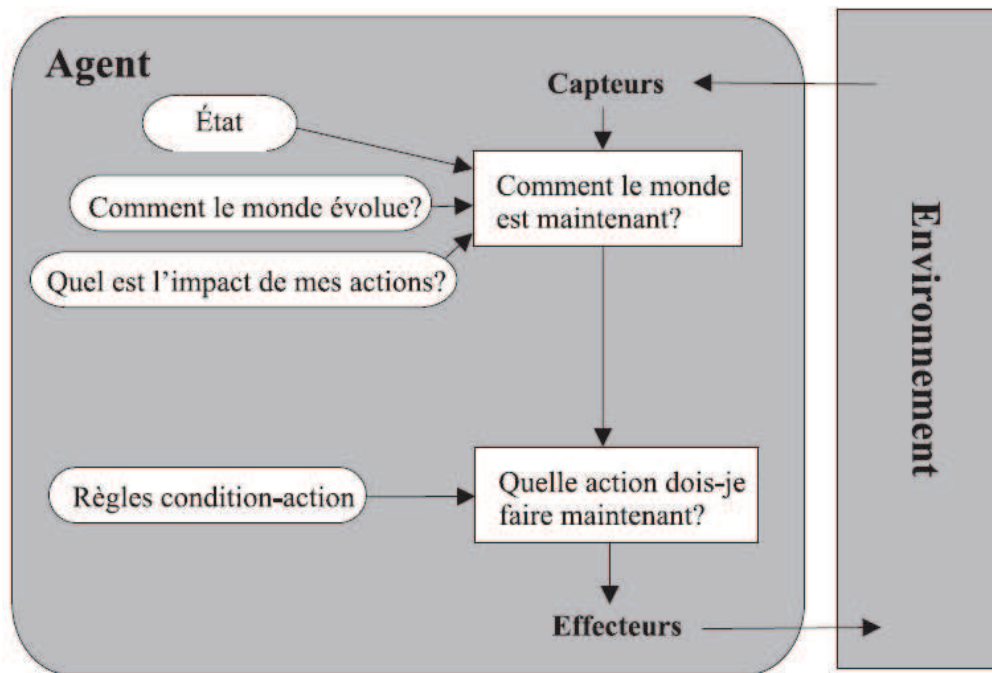


Figure 5.11 : Schéma d'un agent conservant une trace du monde (Russell & Norvig, 1995)

On peut voir, sur la figure, la structure d'un agent conservant une trace du monde. Il utilise ses informations internes (état précédent du monde, l'évolution du monde et l'impact de ses actions) pour mettre à jour ses perceptions actuelles. Par la suite, il choisit ses actions en se basant sur cette perception « améliorée » du monde qui l'entoure.

Les principales inspirations de ce type d'agents sont :

- Insectes sociaux, mammifères sociaux
- fourmis, termites, abeilles, guêpes, loups, rats, primates, oiseaux, poissons
- Eco-systèmes
- Phytosociologie
- Processus physico-chimiques

5.3.2- Les agents délibératifs

Les *agents délibératifs* ont la capacité de résoudre des problèmes complexes. Ils sont ainsi capables de raisonner sur une base de connaissances, de traiter des informations diverses liées au domaine d'application et des informations relatives à la gestion des interactions avec d'autres agents et avec l'environnement (Guessoum, 1996). Ces agents maintiennent une représentation interne de leur monde et un état mental explicite qui peut être modifié par un raisonnement symbolique (Müller, 1996).

Les SMA's délibératifs ont deux problèmes majeurs (Guessoum, 1996) :

- la traduction de l'univers de l'agent en une description symbolique ;
- la représentation symbolique des informations de l'univers complexe des entités et des processus ainsi que la manière de raisonner sur ces informations.

Les agents délibératifs sont des agents qui effectuent une certaine délibération pour choisir leurs actions. Une telle délibération peut se faire en se basant sur les buts de l'agent ou sur une certaine fonction d'utilité. Elle peut prendre la forme d'un plan qui reflète la suite d'actions que l'agent doit effectuer en vue de réaliser son but.

5.3.2.1- Agents ayant des buts

Dans la section précédente, les agents utilisaient leurs connaissances sur l'état actuel de l'environnement pour choisir leurs actions. Toutefois, dans plusieurs situations, cela peut s'avérer insuffisant pour prendre une décision sur l'action à effectuer. Par exemple, on ne peut pas se fier uniquement à l'état actuel de l'environnement pour déterminer la direction que la frégate doit prendre, tout simplement parce que cela dépend aussi de l'endroit où on veut se rendre.

Donc, l'agent a besoin, en plus de la description de l'état actuel de son environnement, de certaines informations décrivant ses buts. Lesquels buts peuvent être vus comme des situations désirables pour l'agent, par exemple, l'arrivée au port d'Alger.

Par la suite, l'agent peut combiner les informations sur ses buts avec les informations sur les résultats de ses actions pour choisir les actions qui vont lui permettre d'atteindre ses buts. Cela peut être facile lorsque le but peut être satisfait en exécutant seulement une action, mais cela peut aussi être beaucoup plus complexe si l'agent doit considérer une longue séquence d'actions avant d'atteindre son but. Dans ce dernier cas, il doit utiliser des techniques de planification pour prévoir les actions devant l'amener à son but.

Contrairement aux agents réactifs, les agents délibératifs, qui raisonnent sur les buts, tiennent compte d'une certaine projection dans le futur. Ils se posent des questions comme « Qu'est-ce qui va arriver si je fais telle ou telle action ? » et « Est-ce que je serai satisfait si cela se produit ? ». Bien entendu, l'agent raisonnant sur ses buts prend, en général, beaucoup plus de temps à agir qu'un agent réactif. Il offre en revanche beaucoup plus de flexibilité. Par exemple, si nous voudrions changer de destination, il faudrait changer toutes les règles de l'agent réactif, tandis que pour l'agent ayant des buts, nous ne changerions que le but.

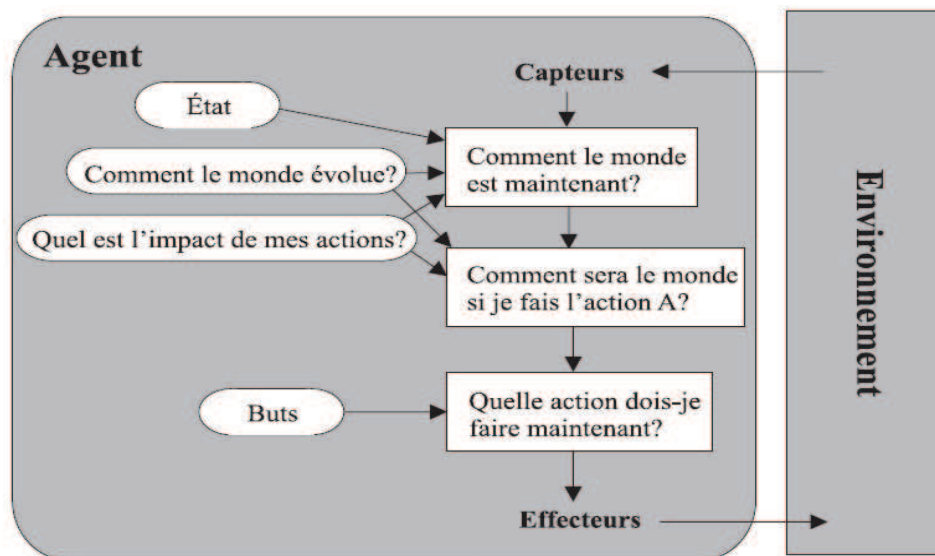


Figure 5.12 : Schéma d'un agent ayant des buts (Russell & Norvig, 1995)

La figure montre la structure d'un agent basé sur les buts. Comme on peut le constater, il est identique à l'agent réactif gardant une trace de l'environnement, sauf qu'il se projette dans le futur pour voir l'impact de ses actions et qu'il choisit ses actions en se basant sur ses buts, contrairement à l'agent réactif qui ne faisait qu'appliquer des règles prédéfinies pour relier ses perceptions à ses actions.

5.3.2.2- Agents utilisant une fonction d'utilité

Dans plusieurs situations, les buts ne sont pas suffisants pour générer un comportement de haute qualité. Par exemple, s'il y a plusieurs chemins possibles pour atteindre le port, certains

seront plus rapides et d'autres plus dangereux. Dans cette situation, l'agent raisonnant uniquement sur ses buts n'a pas de moyens pour choisir le meilleur chemin, son seul but étant de se rendre à destination. Cela se produit car les buts ne procurent qu'une simple distinction entre les états où l'agent est satisfait ou non. En fait, l'agent doit plutôt s'appuyer sur une manière plus fine d'évaluer les états pour être en mesure de reconnaître pour chacun des états son degré de satisfaction. Pour cela, on dit que l'agent va préférer un état à un autre si son utilité est plus grande dans le premier état que dans le deuxième.

Généralement, l'utilité est une fonction qui attribue une valeur numérique à chacun des états. Plus l'état a une grande valeur, plus il est désirable pour l'agent. Dès lors, la spécification d'une fonction d'utilité permet à l'agent de prendre des décisions rationnelles dans deux types de situations où le raisonnement sur les buts échoue. Ainsi, par exemple, lorsqu'il y a des buts conflictuels qui ne peuvent pas être satisfaits en même temps (par exemple, la vitesse et la sécurité), la fonction d'utilité spécifie le compromis approprié entre les différents buts. De même, lorsqu'il y a plusieurs buts possibles, mais qu'aucun d'eux ne peut être atteint avec certitude, la fonction d'utilité permet de pondérer la chance de succès avec l'importance de chacun des buts.

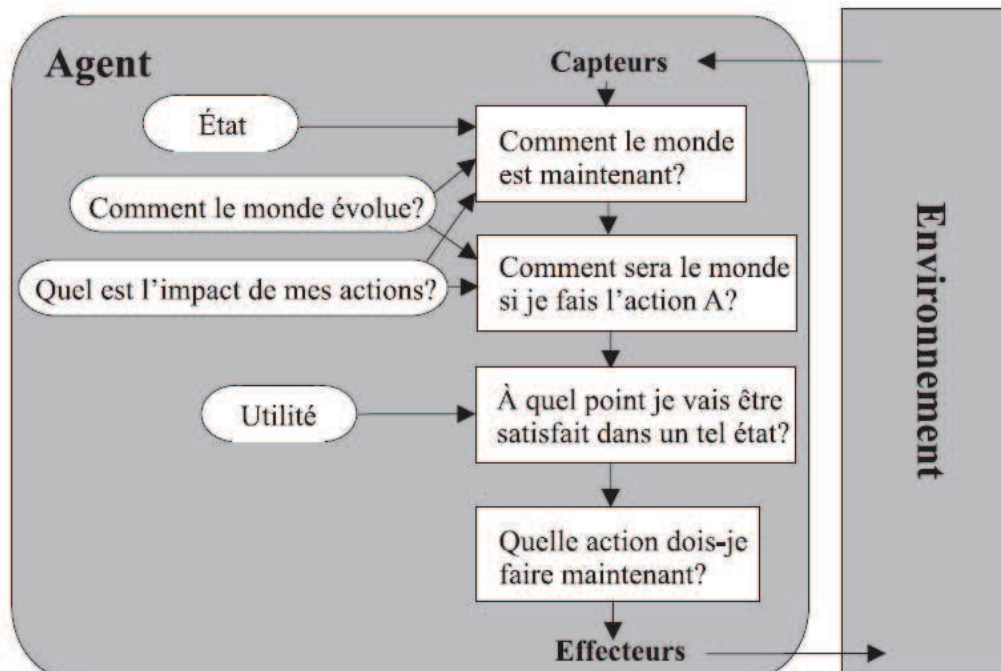


Figure 5.13 : Schéma d'agent basé sur l'utilité (Russell & Norvig, 1995)

La figure montre le schéma d'un agent basé sur l'utilité. On peut voir que l'agent utilise la fonction d'utilité pour évaluer la pertinence d'une action. Il choisit donc les actions qui l'amèneront dans les états ayant la plus grande valeur d'utilité pour lui.

5.3.2.3- Le modèle BDI

L'idée de base de l'approche BDI est de décrire l'état interne d'un agent en termes d'*attitudes mentales* et de définir une architecture de contrôle grâce à laquelle l'agent peut sélectionner le cours d'action de ses attitudes mentales. Il a été défini trois attitudes mentales de base qui sont : les *croyances* (beliefs) les *désirs* (desires) et les *intentions* (intentions). Dans des approches BDI plus pratiques tel que IRMA par exemple (Müller, 1996), il a été montré que ces trois attitudes mentales n'étaient pas suffisantes, une extension a alors été proposée en rajoutant la notion de *buts* (goals) et de *plans* (plans). Nous allons maintenant définir de manière informelle ces différents concepts :

-
- Les ***croyances*** décrivent l'état de l'environnement du point de vue d'un agent (Ferber, 1995). Elles expriment ce que l'agent croit sur l'état courant de son environnement (Müller, 1996).
 - Les ***désirs*** sont une notion abstraite qui spécifie les préférences sur l'état futur de l'environnement d'un agent. Une caractéristique importante des *désirs* est qu'un agent peut avoir des *désirs* inconsistants et qu'il n'a donc pas à croire que ses *désirs* sont réalisables (Müller, 1996).
 - Les ***buts*** représentent les engagements d'un agent pour atteindre un ensemble d'états de l'environnement (Müller, 1996). A partir de la définition précédente, on peut dire qu'un *désir* est une étape dans le processus de création d'un *but*. Si un *désir* d'un agent est poursuivi de manière consistante, il devient l'un des *buts* qui indiquent les options de gestion qu'a un agent (Oliveira, 1998). Cependant, il n'y a, à ce moment là, aucun engagement pour l'exécution de cours d'actions spécifiques. La notion d'engagement d'atteindre un *but* décrit la transition des *buts* aux *intentions*. De plus, il est nécessaire, que l'agent croit que ses *buts* peuvent être atteints (Müller, 1996).
 - Les ***intentions*** représentent les actions que l'agent s'engage à exécuter. A partir du moment où les agents sont limités par leurs ressources, ils peuvent leur arriver de ne pas pouvoir poursuivre tous leurs *buts*. Même si l'ensemble des *buts* créés est consistant, il est nécessaire que l'agent choisisse un certain nombre de *buts* pour lesquels il s'engage. C'est ce processus qui est appelé la formation des *intentions*. Ainsi, les *intentions* courantes d'un agent sont décrites par un ensemble de *buts* sélectionnés avec leur état de traitement (Müller, 1996).
 - Les ***plans*** jouent un rôle très important pour une implémentation pragmatique des *intentions*. Les *intentions* représentent des *plans* partiels d'actions que l'agent s'engage à exécuter pour atteindre ses buts. Par conséquent, il est possible de structurer les intentions en plans plus étendus, et de définir les intentions d'un agent comme les *plans* qui sont couramment adoptés (Müller, 1996).
- Plusieurs applications ont employé ce modèle d'agent, comme par exemple K.Boudaoud dans sa nouvelle approche pour détecter les intrusions dans les réseaux (Boudaoud, 2002)

5.3.3- Les agents hybrides

Les sections précédentes ont présenté deux types d'architectures : réactive et délibérative.

- Les architectures purement réactives ont un comportement assez simpliste,
- Alors que les architectures délibératives utilisent des mécanismes de raisonnement qui ne sont pas faciles à manipuler et qui ne sont pas suffisamment réactifs.

Chacune de ces architectures est appropriée pour un certain type de problème.

Pour la majorité des problèmes cependant, ni une architecture complètement réactive, ni une architecture complètement délibérative n'est appropriée. Comme pour les humains, les agents doivent pouvoir réagir très rapidement dans certaines situations (comportement réflexe), tandis que dans d'autres, ils doivent avoir un comportement plus réfléchi.

Afin d'apporter une réponse à ces imperfections, une architecture conciliant à la fois des aspects réactifs et délibératifs est requise. On parle alors d'architecture hybride, dans laquelle on retrouve généralement plusieurs couches logicielles (Müller, 1996). L'idée principale est de structurer les fonctionnalités d'un agent en deux ou plusieurs couches hiérarchiques qui interagissent entre elles afin d'atteindre un état cohérent de l'agent.

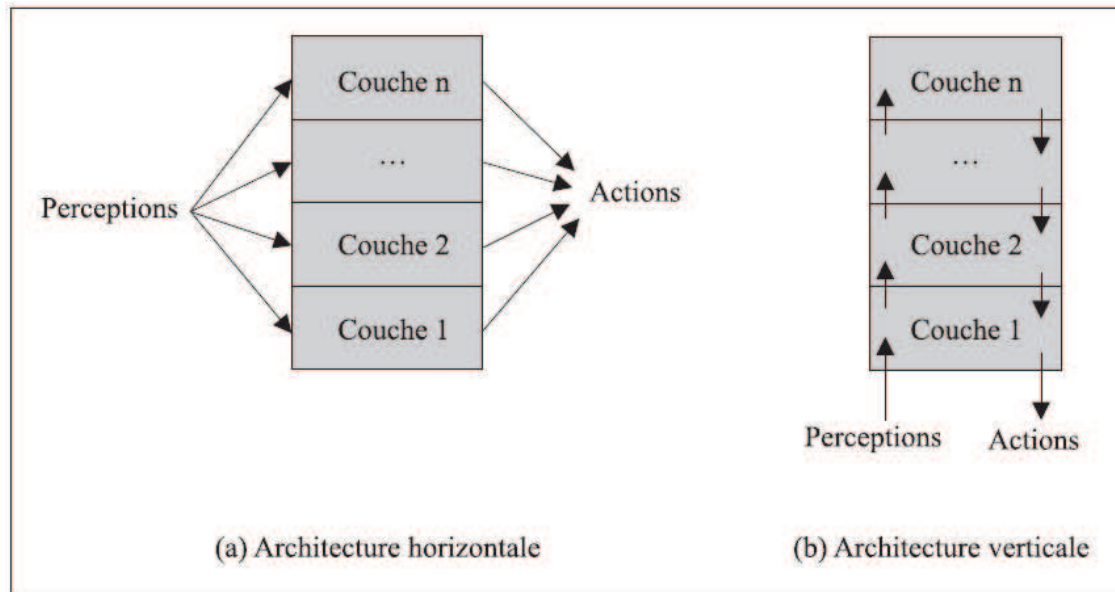


Figure 5.14 : Architectures d'agents en couches (Jennings & all, 1998)

Les couches peuvent être arrangées verticalement (seulement une couche a accès aux capteurs et aux effecteurs) ou horizontalement (toutes les couches ont accès aux entrées et aux sorties), voir figure ci-dessus.

Une approche hybride structurée en couche présente plusieurs avantages (Müller, 1996) :

- Elle permet de modulariser un agent ; ainsi les différentes fonctionnalités sont clairement séparées et reliées par des interfaces bien définies ;
- Elle permet une conception de l'agent plus compacte, ce qui augmente sa robustesse et facilite son "débogage".
- Elle accroît les capacités de l'agent car les différentes couches peuvent s'exécuter en parallèle.
- Elle augmente la réactivité de l'agent car il peut raisonner dans un monde symbolique tout en surveillant son environnement et en réagissant en conséquence.
- Elle réduit les connaissances nécessaires à une couche individuelle pour prendre ses décisions. Par exemple, une couche réactive n'aurait à utiliser que les informations concernant l'état courant de l'environnement alors qu'une couche délibérative aurait à utiliser des informations plus complexes relatives aux attitudes manipulées (croyances, buts, intentions, etc...)

Plusieurs utilisations à base de cette architecture, comme par exemple dans le domaine des transports et la gestion urbaine (Laichour, 2002), (Fayech, 2003) et (M.M.Ould Sidi & all, 2005).

5.4- Apprentissage des agents et des SMA

5.4.1- Apprentissage des Agents

5.4.1.1- Définitions et Différentes formes d'apprentissage

L'apprentissage est un processus d'acquisition de nouvelles compétences. Après apprentissage, un individu ou un groupe d'individus est capable d'un comportement efficace dans des situations qu'il ne savait pas gérer préalablement.

On peut considérer trois modes d'acquisition de compétences :

-
- Par construction ou transmission génétique : l'instinct est un savoir-faire non appris au cours de la vie. Il est inné ou héréditaire. De même un système artificiel peut être conçu doté de certaines comportementales.
 - Par instruction : le sujet apprend à partir d'exemples ou de conseils, il cherche à reproduire le comportement d'un 'modèle' ou d'un 'professeur'.
 - Par expérience : le sujet apprend par processus d'essais d'erreurs.

Philippe Beaune évoque trois aspects d'apprentissage :

- Apprentissage numérique ou symbolique
 - Analyse de données, réseau de neurones, algorithmes génétiques, renforcement...
- Apprentissage déductif ou inductif
 - Reformulation, compilation...
 - Généralisation, classification....
- Classification selon la stratégie
 - Par cœur, par instruction, par analogie, à partir d'exemples, à partir d'observations... (Beaune, 1999)

L'apprentissage est un processus dynamique. D'après G.Clergue, l'apprentissage apparaît comme une trajectoire dans l'espace des phases des systèmes cognitifs du sujet, l'émergence d'un nouveau concept apparaît comme une transition de phase entre un état cognitif ancien et le nouvel état (Clergue, 1997)

L'élaboration d'un modèle d'apprentissage suppose de répondre à certaines questions :

- Qui est en train d'apprendre ?
- Qu'est ce qui est appris : un nouveau concept, un savoir, un savoir-faire, un savoir être?
- Quelles sont les sources de l'apprentissage ? si ce sont des exemples, ceux-ci sont-ils choisis au hasard, sont-ils positifs ou négatifs, simples ou difficiles, comment sont-ils décrits ?
- L'agent (l'apprenant) participe-t-il activement à son apprentissage : peut-il interagir avec son 'professeur' ou d'autres agents (apprenants) peut-il procéder à des expérimentations ?
- La source d'information est-elle bruitée ?
- Dans quel environnement se trouve l'apprenant ?
- De quelles connaissances ou compétences l'agent (l'apprenant) dispose-t-il au départ ?
- Quels sont les méthodes ou outils mis en œuvre pour l'apprentissage ? sont-ils génériques ou spécifiques au domaine, à l'agent (l'apprenant) ?
- Quelles sont les critères et méthodes d'évaluation des résultats de l'apprentissage ?

5.4.1.2- Apprentissage des agents

L'idée derrière l'apprentissage, c'est que les perceptions de l'agent ne devraient pas être utilisées seulement pour choisir des actions, elles devraient être aussi utilisées pour améliorer l'habilité de l'agent à agir dans le futur. L'apprentissage, pour un agent, est très important car c'est ce qui lui permet d'évoluer, de s'adapter et de s'améliorer.

Selon Russel et Norvig dans (Russell & Norvig, 1995), un agent apprenant peut être divisé en quatre composantes, comme le montre la figure ci-dessous :

– Le *critique* indique au module d'apprentissage à quel point l'agent agit bien.

Pour cela, il emploie un standard de performance fixe. Ceci est nécessaire, parce que les perceptions ne fournissent pas d'indications relatives au succès de l'agent. Par exemple, un programme qui joue aux échecs peut percevoir qu'il a mis l'autre joueur échec et mat, mais il a besoin d'un standard de performance pour savoir que c'est une bonne action. Il est important également que le standard de performance soit à l'extérieur de l'agent pour que

l'agent ne puisse pas le modifier dans le but de l'ajuster à son comportement. S'il pouvait le modifier, l'agent ajusterait son standard de performance pour obtenir plus de récompenses pour ses actions, au lieu de modifier ses actions dans le but de s'améliorer.

– Le **module d'apprentissage** utilise une certaine rétroaction sur les actions de l'agent pour déterminer comment le module de performance devrait être modifié pour, on l'espère, s'améliorer dans le futur.

– Le **module de performance** est vu comme étant l'agent au complet lorsqu'il n'y a pas d'apprentissage. C'est-à-dire que ce module peut prendre une des quatre formes que nous avons présentées à la section 5.3 : agent à réflexes simples, agent conservant une trace du monde, agent ayant des buts et agent basé sur l'utilité. Mais, peu importe la manière dont il est construit, son but demeure toujours de choisir des actions à effectuer en se basant sur les perceptions de l'agent.

– Le **générateur de problèmes** donne des suggestions d'actions amenant l'agent à faire de l'exploration. Si on laissait le module de performance choisir tout le temps les actions, il choisirait toujours les meilleures actions selon ce qu'il connaît. Par contre, si l'agent veut explorer un peu, il peut choisir des actions sous-optimales à court terme, mais qui pourraient l'amener à prendre de meilleures décisions à long terme. Par exemple, lors d'une simulation, un agent contrôlant une frégate pourrait suggérer une nouvelle stratégie de défense pour voir si elle est plus efficace que la stratégie actuelle.

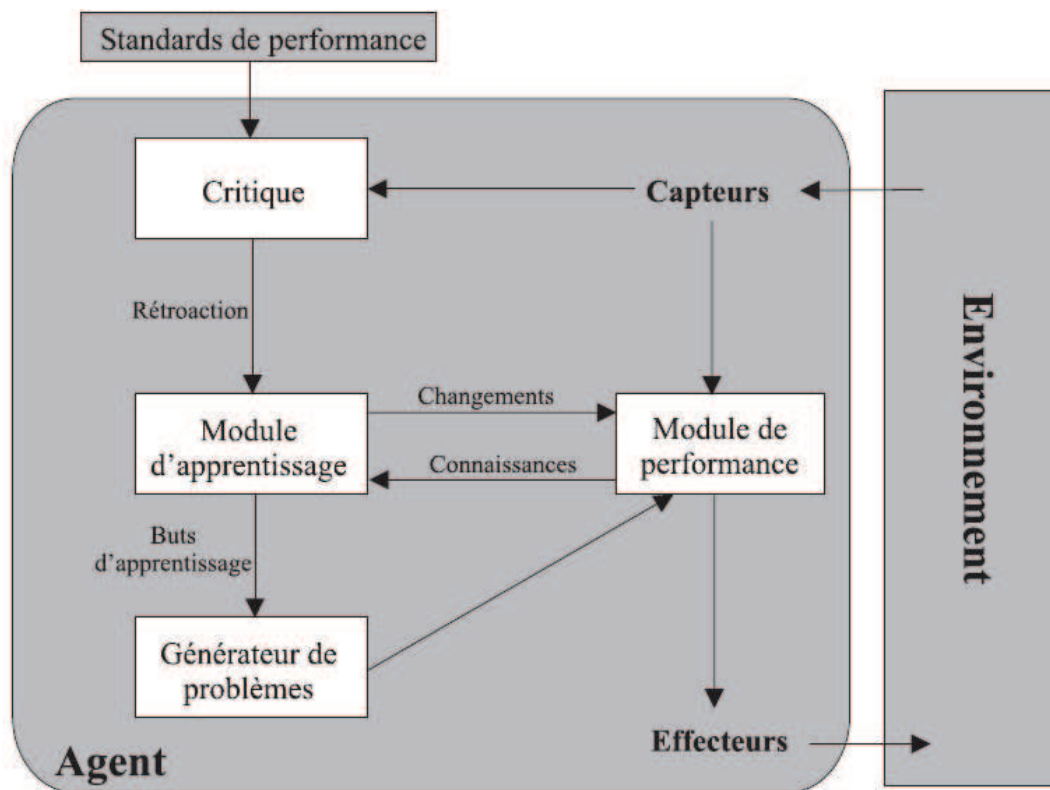


Figure 5.15 : Modèle général d'agent apprenant (Russell & Norvig, 1995)

Illustration pour l'agent taxi apprenant

- Module de performance
 - Connaissances et procédures pour choisir les actions.
- Critique
 - Observe l'agent et donne des informations au module d'apprentissage
- Module d'apprentissage
 - Modifie le module de performance.

- Générateur de problèmes
 - Identifie les possibilités d'amélioration et suggère des expérimentations.

Précédemment, nous avons vu un exemple de réflexe inné chez l'être humain, c'est-à-dire retirer notre main lorsque l'on se brûle, mais les êtres humains peuvent aussi apprendre de nouveaux réflexes. Par exemple, on peut penser à la conduite automobile.

Au début, la conduite est très difficile, car on doit penser à toutes les actions que l'on fait, mais plus on se pratique, moins on réfléchit et plus la conduite devient un réflexe.

Pour les agents, on peut penser à appliquer la même chose. C'est-à-dire, lorsqu'un agent fait face à une situation pour la première fois, il doit délibérer plus longtemps pour choisir ses actions. Mais, avec un module d'apprentissage, plus l'agent effectue des tâches similaires, plus il devient rapide. Son comportement passe graduellement d'un état délibératif, à un état réactif. L'agent a donc appris à exécuter une tâche. D'un point de vue plus technique, on peut dire que l'agent a, en quelque sorte, compilé son raisonnement dans une certaine forme d'ensemble de règles qui lui permettent de diminuer son temps de réflexion. Ce type d'apprentissage peut être très utile pour des agents hybrides.

Ce n'est qu'une façon dont les agents peuvent apprendre, il en existe plusieurs autres. En fait, on considère que toute technique qui permet à un agent d'améliorer son efficacité est une technique d'apprentissage.

5.4.1.2- L'apprentissage par renforcement

Il est tout de même bon de mentionner que la technique la plus utilisée avec les agents est l'apprentissage par renforcement (par essais erreurs) approprié au cas où l'environnement serait inconnu et changeant. Ce type d'apprentissage convient lorsqu'il n'existe pas de modèle ou de professeur capable d'indiquer à l'apprenant ce qu'il doit faire. Cependant l'agent peut apprendre en interagissant avec l'environnement et à partir de son expérience. Pour cela, on donne à l'agent un certain renforcement après chacune de ses actions. Si le renforcement est positif, cela signifie que l'agent a bien agi, celui-ci tentera donc de refaire les mêmes actions.

Si, par contre, le renforcement est négatif, alors l'agent saura qu'il a mal agi et il tentera d'éviter ces actions dans le futur. C'est un apprentissage, par essais et erreurs, qui permet à l'agent de s'améliorer avec le temps. (Dutech & all, 2003)

Bien entendu, pour utiliser ce type d'apprentissage, on doit être dans un environnement autorisant les erreurs. Par exemple, dans le domaine des frégates, on ne peut se permettre d'erreurs, car on ne voudrait pas qu'une frégate reçoive dix missiles avant d'apprendre comment les intercepter adéquatement. Dans ce type d'application, une des solutions est d'utiliser des simulations, procurant ainsi un environnement sécuritaire pour l'apprentissage des agents avant de les implémenter dans le véritable environnement.

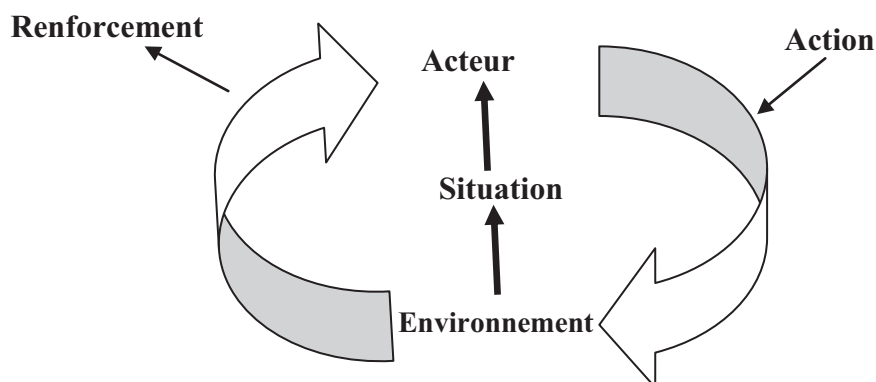


Figure 5.16 : Schéma de principe de l'apprentissage par renforcement

5.4.2- Apprentissage des SMA

L'apprentissage est une composante importante des systèmes multiagents. Ces systèmes évoluent généralement dans des environnements complexes (c'est-à-dire larges, ouverts, dynamiques et non prévisibles) (Sen & Weiss, 1999). Pour de tels environnements, c'est très difficile et même quelque fois impossible de définir correctement et complètement les systèmes à priori, c'est-à-dire lors de la phase de conception, bien avant leur utilisation.

Ceci exigerait de connaître à l'avance toutes les conditions environnementales qui vont survenir dans le futur, quels agents seront disponibles à ce moment et comment les agents disponibles devront réagir et interagir en réponse à ces conditions. Une manière de gérer ces difficultés est de donner à chaque agent l'habileté d'améliorer ses propres performances, ainsi que celles du groupe auquel il appartient.

Il est à noter que l'apprentissage dans un système multiagents comprend l'apprentissage dans un système mono-agent parce qu'un agent peut apprendre en solitaire et de façon complètement indépendante des autres agents. Mais aussi, il l'étend bien au delà dans la mesure où les activités d'apprentissage d'un agent peuvent être influencées considérablement par les autres agents et que plusieurs agents peuvent apprendre de manière distribuée et interactive comme une seule entité cohérente. Dans un environnement multiagents, les agents peuvent apprendre grâce aux autres. (Pesty & all, 2001)

Par exemple, un agent A, qui voudrait savoir comment se rendre à un certain endroit, pourrait demander à un autre agent B s'il connaît un bon chemin. Si B connaît un bon chemin, il peut le transmettre à A, permettant ainsi à l'agent A d'apprendre un bon chemin grâce à l'agent B.

D'un autre côté, les agents peuvent aussi apprendre à propos des autres. Par exemple, un agent peut regarder un autre agent agir dans certaines situations et, à l'aide de ces informations, il pourrait construire un modèle du comportement de l'autre agent. Ce modèle pourrait lui servir pour prédire les actions de l'autre agent dans le futur. Cette information pourrait l'aider à mieux se coordonner ou à mieux collaborer avec l'autre agent.

Une autre facette importante est l'apprentissage des interactions (coordination, collaboration, communication, etc.) entre les agents. Par exemple, si une frégate canadienne doit travailler avec un bateau de guerre anglais, elle doit pouvoir apprendre une manière efficace de se coordonner avec cet allié en tenant compte de ses capacités. En fait, elle doit pouvoir le faire pour toutes les possibilités, c'est-à-dire avec n'importe quel nombre de bateaux provenant de n'importe quel pays et dans n'importe quelle situation. Les agents doivent donc pouvoir adapter leurs mécanismes de coordination et de communication pour que l'ensemble des bateaux puissent agir de manière cohérente et ainsi se défendre le plus efficacement possible. Cette adaptabilité est une forme d'apprentissage de vie sociale.

5.5- Méthodologies de conception d'un SMA

5.5.1- Problématique

- **Problèmes à résoudre :**
 - répartir les tâches et les connaissances
 - coordonner les agents
 - gérer les conflits et le maintien de la cohérence
 - permettre la communication
- **Deux aspects à traiter :**
 - Aspects microscopiques (orientés agent)
 - Comment construire un agent capable d'agir de manière autonome, quelles sont ses représentations et ses comportements ?

-
- Aspects macroscopiques (orientés système)
 - Comment construire une organisation capable d'agir de manière coopérative ?
 - Quels sont ses moyens de communication et de coordination ?

5.5.2- Méthodologie

Afin de guider le processus de conception des SMAs, plusieurs méthodologies de conception ont été proposées. Bien qu'en réalité, il existe très peu de travaux sur ce sujet, nous pouvons en retenir trois : (Gutknecht & Ferber 1999), (Wooldridge, 1999) et (Kinny & Georgeff, 1997).

O. Gutknecht et J. Ferber ont proposé dans (Gutknecht & Ferber 1999) puis évoqué dans (Ferber & all, 2009) et (Ferber & all, 2010), un cadre méthodologique centré sur trois concepts organisationnels :

1. l'**agent** qui est défini comme une entité autonome communicante qui joue des *rôles* au sein de différents *groupes*. L'architecture interne de l'agent n'est pas définie afin de permettre au concepteur de choisir le modèle le plus adapté à son application ;
2. le **groupe** qui est défini comme un moyen d'identifier par regroupement un ensemble d'agents. Un groupe peut être fondé par n'importe quel agent et un agent peut être un membre d'un ou plusieurs groupes ;
3. le **rôle** qui est une représentation abstraite d'une fonction, d'un service ou d'une identification d'un agent au sein d'un groupe. Un rôle peut être attribué à plusieurs agents et un agent peut avoir plusieurs rôles.

Ce modèle est volontairement restreint afin de pouvoir s'intégrer à d'autres méthodologies. Au-dessus de ces trois concepts de base, ils spécifient un modèle structurel, utilisé comme outil de conception, où ils définissent les notions de :

1. **structure de groupe** qui est une description abstraite d'un groupe qui définit les rôles et leurs interactions et qui est représentée par un tuple : $S = \langle R, G, L \rangle$ où :
 - R est l'ensemble des rôles identifiés dans le groupe,
 - G est le graphe d'interactions entre les rôles,
 - L est le langage d'interaction.
2. **structure organisationnelle** qui est l'ensemble des groupes qui définit un modèle d'organisation multi-agents. Elle est représentée par un couple $O = \langle S, Rep \rangle$ où :
 - S est un ensemble de structure de groupes,
 - Rep est le graphe des représentants où chaque arc réunit deux rôles appartenant à deux structures de groupe différentes. Un représentant entre deux structures de groupes est un agent qui aurait simultanément un rôle dans un groupe et un second rôle dans un autre groupe (Gutknecht & Ferber 1999).

M. Wooldridge propose une méthodologie, appelée *Gaia*, qui traite les niveaux *macro* (social) et *micro* (agent) de conception. Tout comme l'approche précédente, la méthodologie est indépendante de l'architecture interne de l'agent. Le processus méthodologique de *Gaia* implique l'utilisation d'un ensemble de modèles définis dans deux phases : *analyse* et *conception* (Wooldridge, 1999).

5.5.2.1- Phase d'analyse

Durant la phase d'analyse, sont définies les entités abstraites qui définissent le modèle organisationnel du système. Ce dernier comprend deux modèles :

1. le **modèle de rôles** qui définit les rôles clés du système. Un rôle est caractérisé par deux types d'attributs :

- les *permissions* ou *droits* identifiant les ressources qui peuvent légitimement être utilisées pour remplir un rôle. Elles définissent également les limites d'utilisation de ces ressources ;
 - les *responsabilités* qui définissent les fonctionnalités d'un rôle.
1. le **modèle d'interactions** qui définit les relations de dépendances entre les différents rôles. Dans ce modèle, pour chaque type d'interaction inter-rôle, est identifié un ensemble de *définitions de protocoles*.

Les définitions de protocoles sont composées des attributs suivants :

- le *but* : une brève description de la nature de l'interaction,
- l'*initiateur* : le ou les rôles responsables de l'initiation de l'interaction,
- le *correspondant* : le ou les rôles avec qui l'initiateur interagit,
- les *entrées* : les informations utilisées par l'initiateur,
- les *sorties* : les informations fournies par le correspondant durant l'interaction,
- le *traitement* : une brève description de tout traitement que l'initiateur du protocole exécute durant l'interaction.

5.5.2.2- Phase de conception

Durant la phase de conception, trois modèles, contenant les entités concrètes du système, sont générés :

1. le **modèle d'agent** qui identifie :
 - Les *types d'agents* qui seront utilisés pour l'implémentation du système,
 - Les *instances d'agents* qui traduiront ces types d'agents à l'exécution ;
2. le **modèle de services** qui définit les principaux services associés à chaque type d'agent. Un service est un bloque cohérent d'activités que l'agent s'engage à accomplir ;
3. le **modèle de communication** qui définit les liens de communication entre les types d'agent. Ce modèle est un simple graphe où les noeuds représentent les types d'agents et les arcs les chemins de communication.

Contrairement à la méthodologie de (Gutknecht & Ferber, 1999), la notion de *structure organisationnelle* n'est pas définie dans Gaia.

L'approche, de D.Kinny et M.Georgeff dans (Kinny & Georgeff, 1997), ne fait pas de distinction entre les phases d'analyse et de conception. Elle propose un ensemble de modèles pour définir d'un *point de vue externe* et d'un *point de vue interne* des systèmes multi-agents construits sur l'*architecture BDI*.

Point de vue externe :

Du point de vue externe, un SMA est défini par deux modèles qui sont indépendants de l'architecture BDI :

1. le **modèle d'agent** qui décrit les relations hiérarchiques entre les différentes classes d'agents abstraites et concrètes. Il identifie également les instances d'agents qui peuvent exister dans le système ainsi que leur multiplicité ;
2. le **modèle d'interactions** qui décrit les responsabilités d'une classe d'agent, les services fournis et les interactions entre les classes d'agents. Ce modèle définit la syntaxe et la sémantique des messages utilisés pour :
 - les communications inter-agents,
 - les communications entre les agents et les autres éléments du système tels que les interfaces utilisateurs.

Point de vue interne :

Du point de vue interne, sont décrites les attitudes mentales de chaque classe d'agent par l'intermédiaire de trois modèles :

-
1. le *modèle de croyances* qui définit les informations sur l'environnement et l'état interne d'un agent et les actions qu'il peut exécuter ;
 2. le *modèle de buts* qui décrit les buts qu'un agent doit atteindre et les événements auxquels il peut répondre ;
 3. le *modèle de plans* qui décrit les plans possibles qu'un agent peut utiliser pour atteindre ses buts.

5.5.2.3- Les étapes de réalisation d'un SMA

La démarche à suivre pour réaliser un SMA est résumée par (Ferber, 1999) :

1. Déterminer :
 - Les agents
 - L'environnement
2. Décrire les lois de l'environnement
3. Identifier les perceptions et les influences (actions) produites par les agents
4. Déterminer les variables internes et capacités des agents
5. Définir les comportements des agents:
 - Si les agents sont cognitifs: décrire la relation entre croyances, buts et actions
 - Si les agents sont réactifs: décrire les stimuli, les tropismes (attraction, répulsion, évitement) ainsi que les tâches (combinaisons d'actions élémentaires)

5.5.3- Plates-formes de développement

Les environnements de développements des SMA, les plus connus sont :

- JADE (Java Agent DEveloppement) est une plate-forme de développement de systèmes multi agents, crée par le laboratoire TILAB et répond aux spécifications FIPA (Foundation for Intelligent Physical Agents).
- ZEUS est un environnement intégré qui utilise la méthodologie Rôle Modeling pour la construction rapide d'applications à base d'agents collaboratifs
- MadKit est une plate-forme multi agents modulaire écrite en Java et construite autour du modèle organisationnel Agent/Groupe/Rôle.
- KQML (KnowledgeQueryandManipulation Language) : Langage de communication entre agents ACL (Agent communication Language).

Et d'autres moins connus :

- ADELPHI, AgentBuilder, Madkit, SamlITalk, LISP, LISP, C++, Prolog, SMALTALK, AgentBuilder, Mask, Mercure, MACE, DIMA, SWORM, Cormas, geamas, Mice, FIPA-OS, LEAP, Actalk, DECAF, JACK, MAGES IV, Magique, Mocah, OSAKA, Pandora II, Alaadin, JAF, Maleva Agent Oriented Programming in Java ,Java Agent Services.
(Boissier & all, 1999)

Les plates-formes fournies comme logiciels libres:

- JADE, MACE, ZEUS, et MADKIT pour les agents cognitifs,
- Et SWORM pour les agents réactifs.

(Guessoum & Occello, 2001) récapitulent les différents environnements de programmation dans le tableau suivant :

Catégorie	Environnements
Outils pour la simulation	Sworm, Cormas, geamas, Mice
Outils pour l'implémentation	Actalk, DECAF, JACK, MAGES IV, Magique, Mocah, OSAKA, Pandora II
Outils pour la conception	Alaadin, JAF, Maleva
Outils pour la conception et l'implémentation	AgentBuilder, DIMA, Mask, Mercure, MACE
Outils pour la conception, l'implémentation et la validation	Jade, Zeus, Madkit, MAST

Tableau 5.2 : Environnements de développement

5.6- Conclusion

L'intelligence artificielle distribuée est née pour résoudre les problèmes de complexité des gros programmes de l'intelligence artificielle, comme par exemple les problèmes de classification : l'exécution est alors distribuée mais le contrôle reste centralisé. Contrairement aux SMA, où chaque agent possède un contrôle total sur son comportement. Pour résoudre un problème complexe, il est plus simple de concevoir des programmes relativement petits (les agents) en interaction, qu'un seul gros programme de rôle similaire. L'autonomie permet au système de s'adapter dynamiquement aux changements imprévus qui interviennent dans l'environnement.

Le fait le plus remarquable dans la montée en puissance de ces nouveaux outils tient sans doute à la transversalité de leur diffusion dans le champ scientifique. En effet, les SMA ont trouvé des terrains d'expérimentation dans de multiples sciences techniques mais aussi dans le champ des sciences de l'Homme et de la Société.

Ainsi, la technologie multi-agent semble être la solution pour le développement des logiciels de demain (Gates, 1999).

Pour nous avons fait appel à cette technologie dans un contexte très pointue de classification automatique de textes et les résultats qui vont être confirmés ultérieurement son bénéfiques.