

# Apprentissage Guidé par les Emotions: Etude d'une Architecture Multi-Agent Apprenante

Jacques Chapelle

► **To cite this version:**

Jacques Chapelle. Apprentissage Guidé par les Emotions: Etude d'une Architecture Multi-Agent Apprenante. 05054, 2005, 13 p. <lirmm-00106695>

**HAL Id: lirmm-00106695**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-00106695>**

Submitted on 16 Oct 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

---

# Apprentissage guidé par les émotions

## Etude d'une architecture multi-agent apprenante.

Jérôme Chapelle\*

\* LIRMM, Université Montpellier II – CNRS  
161 rue ADA – 34392 Montpellier Cedex 5  
chapelle@lirmm.fr

---

*RÉSUMÉ.* Dans ce papier nous présentons une architecture multi-agent permettant à un robot mobile autonome d'apprendre de manière non supervisée. L'apprentissage autonome est réalisé grâce à l'utilisation d'émotions qui représentent les besoins primaires de l'apprenant. Le processus d'apprentissage que nous proposons est inspiré de l'organisation en colonnes corticales du cerveau chez les espèces vivantes. L'architecture multi-agent de type organisationnelle est utilisée pour décrire les interactions entre les entités impliquées dans le processus d'apprentissage.

*ABSTRACT.* In this paper, we present a multi-agent architecture giving a robot the ability to learn in an unsupervised way. An autonomous learning is achieved by using emotions which represent basic needs for the learning entity. The learning process we propose is inspired by the organization in cortical columns and areas of a living being brain. The organizational multi-agent architecture is used to describe the interaction among entities involved in the learning process.

*MOTS-CLÉS :* Systèmes Multi-Agent, Apprentissage, Agent émotionnel, Robotique

*KEYWORDS:* Multi-Agent Systems, Machine Learning, Emotional agent, Robotics

---

-- Numéro de rapport attribué : 05054 --  
Août 2005

## 1. Introduction

Grâce aux progrès de miniaturisation, les robots mobiles disposent de moyens d'interaction avec l'environnement de plus en plus précis et variés. En contrepartie, les structures de contrôle sont de plus en plus complexes à développer et spécifiques à des architectures matérielles données. Il apparaît donc nécessaire de définir une architecture qui génère une structure de contrôle évolutive et adaptative afin que le robot soit capable d'apprendre des tâches de plus en plus complexes, et non connues a priori. Pour réaliser une telle structure et ainsi assurer l'autonomie d'un robot disposant de composants hétérogènes, on se trouve confrontés à des besoins d'apprentissage et d'adaptabilité. Ces problèmes ont d'ailleurs suscité de nombreux travaux en robotique et en informatique ces dernières années. Enfin, la grande diversité de capteurs et d'effecteurs disponibles pour la conception de robots, et la possibilité d'en brancher de nouveaux à un robot existant, soulèvent des problèmes dus à l'hétérogénéité et à l'évolution du nombre de composants qui constituent le robot.

Les travaux antérieurs qui traitent d'apprentissage sont souvent appliqués à des environnements simulés et discrétisés : les mondes grille (Dutech et al., 2001 ; Maçãs, 2001). Ces travaux sont souvent éloignés des problèmes rencontrés en robotique mobile (en particulier pour ce qui est de la perception de l'environnement). De plus, dans beaucoup de travaux, le processus d'apprentissage est supervisé par un observateur externe qui donne à l'entité apprenante une récompense pour motiver le processus d'apprentissage. Pour pouvoir se passer d'un superviseur, il est important de bien définir les motivations qui vont guider le robot dans son processus d'apprentissage. Pour garantir l'autonomie de ce processus, et celle du robot, il faut que ces motivations permettent au robot de percevoir les situations critiques pour lui, ou pour la tâche qu'il doit accomplir. De nombreux travaux (Mataric, 1994b; Gadanho 2002) montrent l'intérêt d'utiliser des émotions, définies comme des motivations très similaires aux besoins primaires auxquels les espèces vivantes sont assujetties, comme la faim ou le besoin de reproduction.

Ainsi, afin de générer des structures de contrôle adaptatives et évolutives, nous proposons dans cet article une architecture multi-agent dont le fonctionnement se base sur le principe du modèle d'interaction cerebro - cérébelleux (Ito, 1984), et utilise des émotions pour guider l'apprentissage.

Ce papier est organisé de la manière suivante. La section 2 décrit le système multi-agent et les raisons qui nous ont conduits à ce choix. La section 3 présente les techniques utilisées pour permettre l'émergence d'une structure de contrôle au sein du système multi-agent. Dans cette section seront introduites les deux phases du processus d'apprentissage : le *babbling learning* et le *satisfying learning*. Puis, dans la section 4 le modèle est évalué ainsi que ses applications potentielles aux problèmes classiques de robotique mobile. Un exemple d'application de cette architecture sur un cas concret sera également présenté dans cette section. Enfin, la section 5 conclut et dresse un bilan des capacités de l'architecture présentée.

## 2. Vers une architecture multi-agent

De tout temps l'homme s'est inspiré de la nature pour concevoir et développer des objets manufacturés. En effet, certains modèles de systèmes multi-agents sont fortement inspirés de l'observation des colonies de fourmis (Ferber, 1999; Drogoul, 1993). Dans notre cas nous allons adopter une approche similaire et prendre pour inspiration le cerveau des êtres vivants, car ceux-ci possèdent les propriétés d'adaptation et d'évolution qui permettent à l'entité apprenante de survivre dans son environnement et d'apprendre de nouvelles tâches de plus en plus complexes. Ainsi seront construits des mécanismes que nous considérons comme importants pour les processus d'apprentissage et d'accumulation de savoir-faire au sein d'une entité autonome évoluant et se déplaçant dans un environnement dynamique.

### 2.1. Les neurosciences

Les systèmes nerveux qui composent les cerveaux des espèces vivantes ont inspiré beaucoup de travaux et de recherches en informatique, en robotique, et en intelligence artificielle (Revel, 1997). En effet, les réseaux de neurones sont très utiles pour résoudre des problèmes jusque là réservés au monde du vivant comme la reconnaissance de forme ou de symboles, ou de la voie humaine. Ces problèmes peuvent être traités grâce à ce genre d'architectures parallèles, qui se prête particulièrement bien aux problèmes d'approximation distribuée et d'apprentissage. Cependant, les réseaux de neurones artificiels classiques semblent trop rigides pour apprendre et mémoriser comme souligné dans (Frankowska et al., 2003), surtout quand on les compare aux performances humaines. Une première explication est que les systèmes nerveux que l'on simule sont des systèmes parallèles alors que l'ordinateur que l'on utilise pour les simuler fonctionne séquentiellement. Une autre difficulté à utiliser des réseaux de neurones réside dans le choix du bon modèle de réseau, et de sa complexité structurelle pour un problème donné. Si le modèle a trop de paramètres, on rencontre un problème *d'overfitting*. Ceci signifie que le réseau de neurones va avoir une erreur de généralisation plus importante que l'erreur empirique, et l'amener à apprendre du bruit, par exemple.

Pour remédier au manque de flexibilité des réseaux de neurones, d'autres travaux (Frankowska et al., 2003) suggèrent de prendre comme point de départ le cerveau des êtres vivants, mais avec une granularité moins fine que le neurone : *les colonnes corticales* (Alexandre, 2001). Cela consiste à regrouper le fonctionnement de plusieurs neurones au sein d'une seule entité apprenante, que nous appelons une colonne corticale. L'équipe CORTEX du Loria utilise un modèle cortical similaire basé sur les concepts d'assemblée neuronale et de colonnes corticales. Chaque colonne réalise des opérations élémentaires, mais d'un niveau fonctionnel plus élevé que le simple neurone formel.

Les colonnes corticales se regroupent alors en aires corticales. Chaque aire corticale est typée par la nature des informations qu'elle traite : aire *sensorielle* ou *motrice*. Des *aires associatives* connectent les aires sensorielles et motrices entre

elles afin de créer des représentations de plus haut niveau et plus structurées. Les *aires frontales* contrôlent et coordonnent le fonctionnement des aires associatives.

## **2.2. Le système multi- agents**

Les propriétés d'adaptation et d'évolution qui sont nécessaires à la réalisation d'une structure de contrôle pour nos robots, nécessitent une architecture qui soit capable d'auto-organisation et qui permette l'émergence de comportements intelligents. De plus notre modèle d'apprentissage requiert la simulation d'un grand nombre d'entités apprenantes, qui ont des buts et des comportements différents, et qui doivent travailler ensemble dans le même environnement. De tels besoins nous ont amenés à considérer les systèmes multi-agents, bien adaptés à la simulation d'un grand nombre d'entités apprenantes autonomes et de leurs interactions. De plus la programmation orientée agent présente des caractéristiques intéressantes :

- Les agents collaborent et se coordonnent afin de produire des performances qualitativement supérieures à celles des unités. De leurs interactions, émergent des comportements complexes.
- Les agents sont autonomes dans leur processus décisionnel (un agent décide par lui-même de l'action à exécuter).
- Des groupes d'agents peuvent être créés, au sein desquels les agents peuvent se rencontrer et échanger des informations. Un groupe est un espace d'interaction dans lequel les agents ont une perception locale de l'environnement dans lequel ils vivent, les communications n'étant possibles qu'entre membres du même groupe.
- Les agents communiquent par échange de messages.

Les caractéristiques que l'on souhaite donner aux entités apprenantes sont proches de celles listées ci-dessus :

- La collaboration et la coordination des colonnes entre elles permettent de produire des performances qualitativement supérieures à celles de chaque colonne : c'est l'intelligence collective. De leurs interactions, émergent une structure de contrôle et des comportements complexes.
- Chaque colonne est autonome : elle peut choisir de filtrer les signaux qu'elle reçoit sur ses entrées et la modulation qu'elle va appliquer à ces informations.
- Les colonnes se rassemblent dans des groupes appelés aires.
- Chaque colonne a une perception locale de l'information. En effet même si une colonne peut avoir beaucoup de connections, elle ne peut avoir une vue complète du système. De plus, une colonne peut envoyer ou recevoir des informations vers ou depuis d'autres colonnes qui sont dans son voisinage, comme des messages échangés entre agents d'un même groupe.

En utilisant la programmation multi-agent, nous pouvons ainsi bénéficier des propriétés d'auto-organisation et d'émergence des systèmes multi-agent pour simuler le processus d'apprentissage réalisé au sein des colonnes et des aires corticales.

### **2.3. Les groupes d'agents : des aires d'apprentissage**

Pour chaque type de perception, on crée une aire corticale sensorielle et pour chaque type d'effecteur, une aire motrice. Une aire associative fait alors la relation entre chaque paire d'aire motrice et sensorielle. Enfin, l'aire frontale contrôle l'activité des aires associatives. C'est cette dernière qui reçoit les émotions et qui doit coordonner l'activité des aires associatives (l'apprentissage et la sélection d'actions). Une aire d'apprentissage est un espace d'interaction privilégié qui remplit une certaine fonction, et au sein de laquelle les colonnes échangent des informations. De la même manière, un groupe est un espace d'interaction privilégié qui réalise un comportement, et au sein duquel les agents interagissent par envoi de messages.

## **3. Le processus d'apprentissage**

### **3.1. Les émotions : motivations de l'apprentissage**

Quand on réalise un système apprenant, on cherche à obtenir un système qui choisisse par lui-même la politique qui lui permettra de maximiser les récompenses qu'il obtient. Dans le cas d'un apprentissage supervisé, le professeur observe l'élève, et estime l'efficacité des actions qu'il fait. Le but de l'apprenant étant de produire les comportements qui satisferont le professeur. De nombreux travaux sur les émotions ont ainsi vu le jour dans divers domaines comme en neurosciences (Damasio, 1994), en informatique (Minsky, 1985 ; Shibata et al., 1996) ou en psychologie. Ces recherches ont également donné lieu à des expérimentations comme pour le modèle DARE (Maçãs, 2001) et à des applications commerciales avec le chien Aibo (Sony).

Dans un système apprenant autonome (non supervisé), il faut donc trouver les motivations qui vont permettre de structurer l'apprentissage. Ceci afin de permettre au système de focaliser son attention, et donc l'apprentissage, sur une partie du problème. Dans notre travail, nous cherchons les motivations dont l'entité apprenante a besoin pour apprendre par elle-même et développer ses connaissances, afin que l'intelligence émerge de motivations relativement simples, mais qui ont une signification forte pour l'entité apprenante. Pour arriver à cela, il est nécessaire que l'entité vive suffisamment longtemps, et rencontre beaucoup de situations différentes pour qu'elle développe ses connaissances. En conséquence nous devons lui fournir les motivations qui vont lui permettre de continuer à évoluer, tout en respectant les contraintes de l'environnement et de son propre corps. Par exemple, le jeu peut être calculé par une estimation de l'énergie consommée pour réaliser des mouvements ; pour la faim, la valeur peut être calculée en considérant le niveau de charge des batteries. Le comportement attendu est que le robot joue jusqu'à ce que ses besoins énergétiques deviennent critiques, le forçant à rentrer à la base pour se recharger.

Pour calculer les émotions nous nous sommes inspirés du modèle de Satisfaction-Altruisme (Simonin, 2001), qui est une extension des progress estimators de M.J.

Mataric (Mataric, 1994a). Comme pour le modèle Satisfaction-Altruisme, nous distinguons deux types de motivations relatives à :

- l'état interne, qui est une évaluation des valeurs critiques de certaines perceptions internes : santé, intégrité physique, faim, besoin de jouer, curiosité.
- l'état externe, qui est une composition des niveaux de satisfaction donnés par les agents se trouvant dans le voisinage de l'entité apprenante. Ceci afin de réaliser un apprentissage social. Des travaux qui parlent d'apprentissage social ont montré qu'une architecture où les agents apprennent à satisfaire leurs voisins permet d'obtenir une meilleure coopération (Chapelle et al., 2002; Mataric, 1994a).

Dans le modèle Satisfaction-Altruisme, la satisfaction personnelle est définie comme une valeur constamment calculée qui représente, pour l'action entreprise, la distance au but. Dans notre modèle, pour chaque émotion, une valeur positive représente un but à atteindre, et une valeur négative une gêne à éviter.

Ainsi, toutes les différentes valeurs de satisfaction d'un agent sont combinées afin de calculer l'état émotionnel de celui-ci. Pour permettre un apprentissage social, l'état émotionnel de l'agent est alors transmis aux agents se trouvant dans son voisinage. Cet état émotionnel est alors utilisé par les agents qui le perçoivent comme une motivation à satisfaire de sorte qu'ils apprennent également à satisfaire leurs voisins.

### **3.2. Le processus d'apprentissage**

Comme nous l'avons vu précédemment, la structure de contrôle du robot doit être adaptée à sa structure physique, et doit évoluer pour apprendre et réaliser des tâches de plus en plus complexes. C'est pourquoi l'apprentissage au sein du système multi-agent est réalisé en deux phases : la première, le *babbling learning*, consiste à regrouper les capteurs et les effecteurs similaires (dans le sens où ils sont physiquement proches, ou s'ils ont le même mode de fonctionnement). La deuxième phase, le *satisfying learning* utilise des groupements déjà créés, notamment pendant la phase de babbling, comme base pour construire des connaissances plus abstraites.

#### **3.2.1. La première phase : le babbling learning**

Le babillage consiste, pour le bébé, à produire des sons, d'abord de manière accidentelle, puis à en reproduire certains de manière régulière et répétée. A la fin de la première année, le babillage devient plus clair et quelques sonorités sont répétées intentionnellement, l'enfant a alors les capacités à prononcer ses premiers mots.

La phase de *babbling learning* consiste donc à générer des mouvements aléatoires et à observer leurs impacts sur les perceptions, comme mentionné dans (Dufossé). Pour réaliser ceci, un gestionnaire envoie des valeurs aléatoires aux effecteurs, et observe leurs impacts sur l'environnement en écoutant les messages envoyés par les agents capteurs. On associe, par ce processus, le vecteur des mouvements perçus en entrée au vecteur des commandes motrices générées. Ainsi, quand le gestionnaire remarque une similarité de fonctionnement, il crée un nouveau

groupe correspondant. Un *représentant* de groupe est alors placé dans celui-ci dans le but d'interagir avec les couches plus hautes en utilisant un nombre limité d'ordres synthétisés (par exemple avancer, reculer, tourner à droite, à gauche).

A la fin de cette phase de babillage on obtient une structure de contrôle proche du schéma sensori-moteur du robot et qui prend en compte les contraintes physiques et la dynamique du système.

### 3.2.2. La deuxième phase : le *satisfying learning*

Dans cette phase, on utilise les abstractions des couches physiques faites pendant la phase de babbling. Cette phase de construction des connaissances est réalisée en essayant de satisfaire les émotions ressenties. Quand une nouvelle procédure est apprise, elle est ensuite considérée comme une primitive qui peut être utilisée pour la construction d'autres procédures plus abstraites. Pour réaliser cette phase d'apprentissage il est nécessaire de définir les trois types d'agent suivants :

Le système dispose d'un *gestionnaire d'émotions* qui perçoit les émotions qui sont calculées à partir d'informations récupérées du système comme la capacité de charge des batteries qui donne l'émotion de faim. Le gestionnaire d'émotions va agir dans le but de satisfaire les émotions qu'il ressent. Pour ce faire il crée dans son groupe un délégué pour chaque émotion. Il peut alors apprendre à focaliser son attention sur la gestion d'une de ses émotions en utilisant une méthode d'apprentissage par renforcement où la récompense est donnée par la valeur moyenne des émotions.

Chacun des *délégués* cherche à satisfaire l'émotion à laquelle il est assigné. Il va donc chercher les situations qui lui permettent de se satisfaire. Quand le délégué remarque qu'un certain type de perception lui amène du plaisir, il mémorise cette configuration. D'un point de vue du robot, une configuration capteur de ce type représente un but intermédiaire à atteindre pour satisfaire son émotion. De la même manière, une configuration moteur amenant du plaisir correspond à l'action à entreprendre pour obtenir satisfaction. Pour mémoriser une configuration, le délégué crée un groupe avec les agents concernés, et enregistre les valeurs de chacun d'entre eux. Au fur et à mesure que le robot évolue et se déplace, il va apprendre les objectifs intermédiaires, et les actions à entreprendre, qui lui amènent satisfaction.

Les *représentants* sont les premiers intermédiaires entre la couche physique et la couche émotionnelle. Les premiers représentants sont créés durant la phase de babbling pour synthétiser les ordres que peuvent émettre ou recevoir les capteurs ou effecteurs qu'ils représentent. De la même manière, un représentant créé durant la phase de *satisfying* propose à ses représentants des synthèses de ses délégués.

Un représentant synthétise les ordres que peuvent recevoir ou envoyer ses délégués. Pour une valeur numérique, il discrétise celle-ci à l'aide d'une fonction de transfert afin de proposer un nombre fini d'états. Par exemple, pour un moteur : arrêté, lent, rapide.. Quand il dispose de plusieurs délégués, il peut proposer différents couplages. Soit en générant les combinaisons (produit cartésien) des



ordres envoyés ou reçus : un représentant A qui possède deux délégués B et C pourra proposer les ordres données par: Ordres de A = Ordres de B  $\times$  Ordres de C. Soit en proposant des relations entre différents ordres :  $v_A = \alpha.v_B + (1 - \alpha).v_C$ .

Un représentant affine et propose de nouvelles synthèses : en proposant une discrétisation plus fine des valeurs, par exemple : très peu, peu, moyen, ou en ajustant les différents coefficients qui entrent en jeu dans les fonctions de transfert et les relations de valeurs. Lorsque les délégués dont il dispose ne lui permettent pas d'identifier une situation ou de réaliser une action, le représentant peut aller chercher parmi les délégués de ses délégués, celui qui correspond à ses attentes.

### 3.3. Les outils d'apprentissage

#### 3.3.1. L'apprentissage par renforcement

Les algorithmes d'apprentissage par renforcement (Sutton and Barto, 1998) sont très bien adaptés aux problèmes de sélection d'actions. La simplicité de ces algorithmes et leur utilisation possible dans des environnements non déterministes les ont rendu populaire et référencés dans de nombreux travaux (Parker, 1994; Shibata et al., 1996). Dans cette méthode, l'agent apprend grâce à des récompenses externes reçues de l'environnement. La récompense est interprétée comme un renforcement scalaire positif ou négatif. L'objectif du système est d'améliorer son processus de sélection d'action afin de maximiser les récompenses externes reçues. Le principal problème de cette approche est l'évaluation des récompenses. Dans des environnements grille simulés, les valeurs de récompense peuvent être facilement calculées (Dutech et al., 2001; Boutillier, 1999). Dans des systèmes réels, les agents doivent percevoir par eux-mêmes la réussite ou l'échec de leurs actions. Dans notre approche émotionnelle la réussite ou l'échec est donné par la valeur de l'émotion qui est traitée.

Pour permettre un tel apprentissage, les comportements de l'agent sont définis par des règles pondérées, de la forme :  $(p_i, a_i) \Rightarrow (w_i, t_i)$ . Pour une perception donnée  $p$ , la probabilité que l'action  $a$  soit choisie est proportionnelle à la récompense  $w$  qu'il a reçu dans le passé, et inversement proportionnelle au nombre d'essais  $t$ . Cette pondération assure l'exploration de tous les comportements possibles.

Ces règles sont donc utilisées dans un processus de sélection d'action qui s'apparente au modèle EMF introduit par (Drogoul, 1993).

#### 3.3.2. Discrétisation des valeurs : une approche basée sur la logique floue

Les algorithmes d'apprentissage par renforcement gèrent des règles dans lesquelles pour une perception donnée  $p$ , une action  $a$  est déclenchée avec une probabilité  $w$ . Ce modèle est bien adapté pour sélectionner un comportement plutôt qu'un autre, mais il ne peut s'appliquer à un modèle où les perceptions et les actions sont exprimés avec des valeurs continues (dans notre modèle ces valeurs

appartiennent à  $[0;1]$ ). Chaque perception ou action peut prendre une infinité de valeurs. Prendre chacune de ces valeurs pour construire des règles amènerait à un nombre infini de règles. Pour éviter cette explosion combinatoire, nous nous sommes inspirés de la théorie des sous-ensembles flous (Zadeh, 1965). Nous avons donc décidé de ne considérer qu'un nombre fini d'états donnés par une partition de l'espace des états. Au lieu d'un ensemble infini de valeurs, une perception (ou une action) peut être exprimée en utilisant des états comme "peu", "moyen" ou "beaucoup". La correspondance entre l'espace de valeurs continues et l'ensemble d'états finis est faite en utilisant une fonction de transfert.

Grâce à ce genre de discrétisation des perceptions et des actions, l'utilisation d'une méthode d'apprentissage par renforcement devient possible car il y a un nombre fini de perceptions et d'actions.

#### **4. Application, Discussion**

Nous avons réalisé des expérimentations pour montrer que le robot développe et améliore des connaissances et des savoir-faire. Ceci afin de montrer les propriétés du modèle : les capacités d'évolution et d'adaptation. Dans ces expérimentations, un robot mobile est placé dans un environnement dans lequel se trouvent une base de recharge, des murs, et des objets qui peuvent être déplacés. Le robot perçoit trois émotions : la faim, le besoin de jouer, l'intégrité physique, et doit gérer les priorités à satisfaire chacune d'elle.

##### **4.1. Le simulateur**

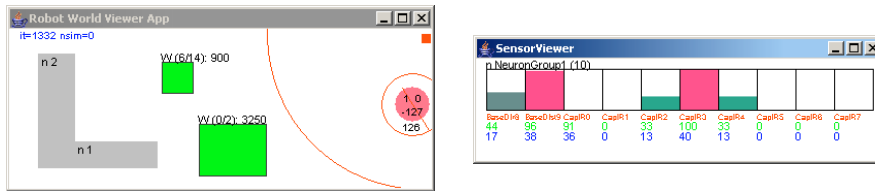
Cette architecture présente l'intérêt principal d'être applicable à des robots réels. Cependant utiliser de vrais robots pour tester notre modèle d'apprentissage amène plusieurs problèmes : tout d'abord les robots coûtent cher, et réaliser des simulations avec un grand nombre de robots est donc difficile. Un autre problème vient des batteries des robots qui prennent longtemps à recharger, ce qui interrompt les expérimentations pendant de longs moments. Enfin, l'expérimentation sur des robots réels rends difficile l'observation et le debuggage du processus d'apprentissage.

C'est pour ces raisons que nous avons choisi de simuler des robots existants grâce à un simulateur multirobots (Simonin et al., 2002) qui s'exécute sur la plateforme multi-agent MadKit (Gutknecht et al., 2000). La structure et les caractéristiques mécaniques des robots sont les mêmes que celles des robots développés dans le cadre du projet CoRoM. Ceux-ci ont déjà été utilisés dans d'autres travaux (Simonin, 2001; Lucidarme et al. 2002). Dans le simulateur, chaque capteur et effecteur est simulé avec précision, et un bruit aléatoire est même appliqué aux perceptions, ce qui est très important pour nous afin d'obtenir des simulations réalistes, et crédibles pour des roboticiens. De plus, l'utilisation du simulateur nous

permet, pendant la simulation, d'observer les valeurs des capteurs, et de modifier les valeurs des effecteurs et des émotions.

#### 4.2. Une expérimentation

Dans cette section, nous allons présenter un exemple d'application de notre architecture sur un robot de type khepera. L'accumulation de savoir-faire se fait par exercices de difficulté croissante. Sur les Figures 4a et 4b nous présentons un exemple où un robot est équipé de huit capteurs et émetteurs infrarouges à sa périphérie lui permettant de détecter les obstacles sur le principe du radar.

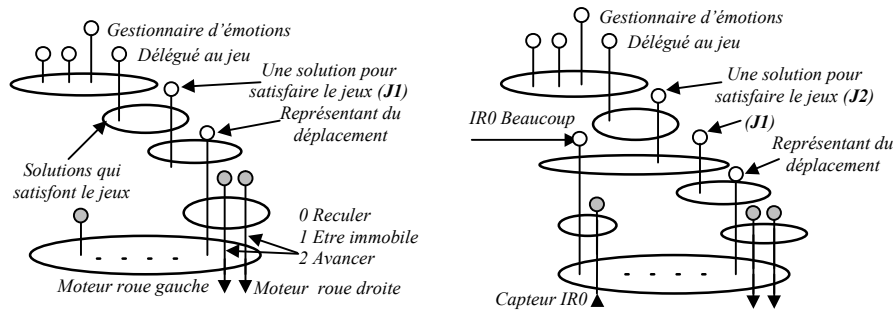


**Figures 4a et 4b.** *A gauche, capture d'écran d'une simulation en cours. A droite, les valeurs des capteurs au même moment : on peut voir la réaction des capteurs infrarouges à la détection du mur.*

Le robot dispose également de capteurs de contact pour pouvoir détecter s'il touche un objet : une collision ayant un impact négatif sur l'émotion qui représente l'intégrité physique. La base grâce à laquelle le robot peut recharger ses batteries, émet un signal radio que le robot perçoit sous la forme de deux valeurs : distance et direction. Quand le robot est situé à proximité de la base, c'est à dire dans la zone délimitée par l'arc de cercle sur la figure 4a, ses batteries se rechargent et il ressent un plaisir de satiété (satisfaction de la faim). Il perçoit également une émotion de jeu, qui évolue proportionnellement à la vitesse de déplacement du robot. L'expérimentation débute avec un robot dont le cerveau est « vierge ». La première phase d'apprentissage est donc la phase de *babbling learning*. Le robot actionne ses effecteurs, et parmi ceux-ci remarque que les deux moteurs des roues ont un impact sur le capteur qui donne la direction de la base, et sur celui qui représente la distance à la base (chacun des moteurs pouvant tourner dans un sens ou dans l'autre). Il crée donc un groupe contenant les deux agents moteurs, ainsi qu'un représentant qui sera responsable de proposer les différentes combinaisons motrices intéressantes, parmi lesquelles : avancer, tourner, faire un demi tour, ne pas bouger.

Une fois la phase de babbling terminée, le gestionnaire d'émotions va essayer des actions parmi celles disponibles, et observer l'impact sur les émotions. Les actions vont être choisies parmi celles proposées par les délégués aux émotions, puis par les représentants de groupe d'effecteurs, comme le représentant du déplacement créé durant la phase de babbling, et enfin par les actions possibles sur tous les effecteurs.

Il commence donc par essayer les représentants déjà existants. Le délégué au jeu remarque alors que parmi les actions proposées par le représentant du déplacement, beaucoup satisfont le jeu, à l'exception de l'action « ne pas bouger ». Un groupe est alors créé par le délégué au jeu, et un nouveau représentant y est placé afin de proposer les actions connues pour satisfaire la faim. Ce représentant, noté *J1* sur la figure 5a ci-dessous, est donc une solution au problème de jeux.



**Figures 5a et 5b.** Organisation de la structure de contrôle. A gauche : apprendre à jouer. A droite : anticiper un obstacle.

Ce premier savoir-faire acquis, on augmente la difficulté du problème et on place le robot dans un environnement contenant des obstacles. Pour assouvir son besoin de jeux, le robot effectue des déplacements, et entre alors en collision avec un mur. Le délégué à l'intégrité physique perçoit alors une chute de l'émotion qu'il représente. Il cherche alors parmi les actions possibles celles qui vont résoudre son problème, et trouve dans les neuf actions proposées par le représentant du déplacement, un sous ensemble d'actions qui permet au robot de s'éloigner de l'obstacle. Le délégué à l'intégrité physique crée alors un groupe dans lequel un nouveau représentant est placé afin de pouvoir résoudre ce problème par la suite.

Le robot se remet alors à jouer, et rencontre d'autres murs. Le gestionnaire d'émotion remarque que quand le robot joue, l'intégrité physique peut chuter indépendamment de l'action entreprise précédemment. Il cherche donc parmi les perceptions celles qui interviennent à chaque fois qu'il rencontre ce problème. Il remarque que la chute d'intégrité physique coïncide avec la valeur « beaucoup » du capteur infrarouge situé devant lui. Cette perception représente donc une exception à prendre en compte dans la procédure de jeux *J1* apprise précédemment. Le gestionnaire de jeux crée donc un nouveau représentant *J2*, comme sur la figure 5b. Ce dernier fait appel *J1* sauf quand un obstacle est détecté. Dans ce cas, *J2*, va apprendre à sélectionner, parmi les autres actions possibles, celle qui permet d'éviter, et donc d'anticiper, la collision.

## 5. Conclusion

Nous avons analysé les besoins d'adaptabilité et d'évolution des structures de contrôle des robots mobiles. Ces besoins découlent de l'hétérogénéité des composants physiques, et d'une nécessité de tolérance à l'évolution de l'environnement. Ces besoins d'adaptation et d'évolution nous ont amené à nous inspirer des systèmes nerveux existants, et à considérer les systèmes multi-agents pour leur capacités à simuler des organisations complexes et les interactions entre les entités qui les composent. Afin de permettre une autonomie dans ce processus de construction de savoir-faire, nous avons souligné l'intérêt de définir des motivations qui représentent des besoins primaires pour la survie du robot, et pour la réalisation de tâches plus complexes.

Nous avons montré les capacités du modèle à permettre à l'apprenant de se construire une représentation du monde qui lui est propre, et qui tient compte de sa structure physique et des émotions qu'il doit satisfaire. De plus, la structure de contrôle du robot se construit progressivement de manière à permettre l'apprentissage de tâches de plus en plus complexes. Les propriétés d'adaptation et d'évolution rendent ainsi possible l'ajout de nouveau composants physiques à la structure du robot existant sans avoir à reconstruire toutes les connaissances : il suffit simplement de faire une phase de babbling learning. De la même manière, la détérioration ou la destruction d'un capteur peut se produire sans que le système ne soit bloqué. En effet, en cas de composant défaillant, le système n'arrivant plus à résoudre ses problèmes, va explorer de nouvelles possibilités d'action ou de perception et focaliser son attention sur celles qui lui permettront d'assouvir ses émotions. On peut observer ce phénomène sur des personnes aveugles qui apprennent à mieux utiliser leur ouïe.

Dans les premières expérimentations que nous avons réalisé, nous avons observé l'émergence de certains comportements connus comme l'évitement d'obstacle, présenté dans ce papier, ou la recherche de la base de recharge quand le besoin se fait ressentir. Ces résultats préliminaires nous encouragent à continuer le développement de notre simulateur afin de réaliser d'autres expérimentations dans des environnements variés et sur des robots d'architecture différente. Nous prévoyons également d'utiliser l'émission de l'état émotionnel du robot afin de réaliser un apprentissage social comme cela avait été fait dans des travaux antérieurs (Chapelle et al., 2002) utilisant une architecture plus simple.

## 6. Bibliographie

Alexandre F., Un survol de la modélisation corticale, CORTEX Team, 2001.

Boutilier C., Sequential Optimality and Coordination in Multiagent Systems. *In Proceedings of the 16th Int. Joint Conference on AI*, 1999.

- Chapelle J., Simonin O., Ferber J., How Situated Agents can Learn to Cooperate by Monitoring their Neighbor's Satisfaction. *ECAI'2002*, Lyon, France, 2002.
- Damasio A.R., *Descartes' error – Emotion, reason and human brain*. Picador, London, 1994.
- Drogoul A., De la simulation multi-agents à la résolution collective de problèmes. PhD Thesis University of Paris VI, 1993.
- Dufossé M., Kaladjan A., Frolov A. A., Bensmail S., Ouezdou F. B., Adaptive Neural Network Control System inspired from the Cerebro-Cerebellar Network for Manipulation.
- Dutech A., Buffet O., Charpillet F., Multi-Agent Systems by Incremental Gradient Learning. *International Joint Conference on AI*, 2001.
- Jacques Ferber. *Multi-Agent Systems: Introduction to Distributed Artificial Intelligence*. Addison Wesley Professional, 1999.
- Frankowska H., Burnod Y., Dufossé M., Modeling the Satisficing-Learning in Basal Ganglia: a Viability Approach, 2003.
- Gutknecht O., Ferber J., Michel F., Une expérience d'architecture de plateforme multi-agent générique. *JFIAD 2000* p224
- Ito M., *The Cerebellum and neural control*. Raven Press, New York, 1984.
- Lucidarme P., Simonin O. and Liegeois A., Implementation and Evaluation of a Satisfaction-Altruism Based Architecture for Multi-Robot Systems. *In ICRA*, 2002.
- Maçãs M., Ventura R., Custódio L., Pinto-Ferreira C., Experiments with an emotion based agent using the DARE architecture. *Proceedings of the Symposium on Emotion, Cognition, and Affective Computing (AISB'01 Convention)*, UK, 2001.
- Mataric M.J., Learning to Behave Socially. *From animals to animats 3*. April 1994.
- Mataric.M.J., Reward functions for accelerated learning. *In Proc. of 11<sup>th</sup> ICML*, 1994.
- Minsky M., *The Society of Mind*. New York : Simon & Schuster, New York, 1985.
- Parker L., ALLIANCE: An architecture for fault tolerant, cooperative control of heterogeneous mobile robots. *Proceedings of the IIROS*, 776-783, 1994.
- Revel A., Contrôle d'un robot autonome par approche neuro-mimétique. PhD Thesis ENSEA Cergy Pontoise, France, 1997.
- Shibata T., Ohkawa K., Tanie K., Spontaneous behavior of robots for cooperation - emotionally intelligent robot system -. *Proceedings of the IEEE ICRA*, 2426-2431, 1996.
- Simonin O., Le modèle satisfaction-altruisme : coopération et résolution de conflits entre agents réactifs, application à la robotique. PhD thesis, UM2 2001.
- Simonin O., Michel F., Chapelle J., Ferber J.. Un simulateur de systèmes multi robots dans MADKIT. *JFIADSMA 2002*, Lille, France.
- Sutton R.S., Barto A.G., *Reinforcement Learning: an Introduction*. MIT Press, Cambridge, MA, 1998.
- Zadeh L.A., Fuzzy Sets. *Information and Control*, volume 3, n°8, pp 338-353, 1965.