

Intelligence Artificielle Développementale Appliquée aux Interactions Sociales dans un Système Multi-Agents

Isabelle FLORES, Alix GONNOT

Février 2016

Résumé

L'objectif de ce projet est d'appliquer les principes de l'intelligence artificielle développementale dans un système multi-agents. Dans ce but, nous avons travaillé avec un agent artificiel capable d'apprendre le fonctionnement du monde autour de lui, et nous l'avons fait évoluer dans un environnement où il n'est plus seul mais en compagnie d'un autre agent. L'objectif est ainsi d'ajouter à son comportement une dimension sociale et d'observer les effets de ce comportement social sur ses capacités d'apprentissage.

Mots-clés

Intelligence Artificielle Développementale, Systèmes Multi-Agents, Comportement Social

1 Introduction

Dans le cadre de leur formation, les étudiants de première année de Master Informatique sont amenés à réaliser un projet de recherche dans le but de découvrir le monde de la recherche scientifique.

Notre projet de recherche nous a été proposé par Mme Véronique Deslandres, membre de l'équipe SMA (Systèmes Multi-Agents), faisant elle-même partie du LIRIS (Laboratoire d'InfoRmatique en Image et Systèmes d'information), qui nous a encadrées au cours de sa réalisation.

Ce projet a pour objectif de faire le lien entre intelligence artificielle développementale et systèmes multi-agents. Dans ce but, nous avons travaillé avec Ernest, un agent artificiel fonctionnant selon les principes de l'intelligence artificielle développementale et capable d'apprendre à évoluer dans son environnement. Nous avons fait interagir deux agents de type Ernest dans l'optique de leur apprendre à développer un comportement social et d'observer comment ce dernier influe sur leur apprentissage de l'environnement.

La base de code que nous avons utilisée a été fournie par M. Simon Gay, membre de l'équipe TWEAK (Traces, Web, Education, Adaptation, Knowledge), qui a beaucoup travaillé sur l'agent Ernest puisque celui-ci était au cœur de son sujet de thèse [4]. Il nous a assisté tout au long du projet pour apporter à son travail les modifications nécessaires à nos expérimentations.

2 Contexte

2.1 Etat de l'art

2.1.1 L'Intelligence artificielle développementale

Une définition de l'intelligence artificielle, attribuée à Marvin Lee Minsky, est par exemple : "*The building of computer programs which perform tasks which are, for the moment, performed in a more satisfactory way by humans because they require high level mental processes such as : perception learning, memory organization and critical reasoning*".

Cependant, c'est loin d'être la seule définition de l'intelligence artificielle existante car il s'agit d'un domaine extrêmement vaste et qui a de nombreuses applications. Cette variété impose une grande diversité des pratiques et surtout des définitions. Les chercheurs travaillant dans ce domaine n'hésitent d'ailleurs pas, à l'image d'Hans Moravec [3], à chercher l'inspiration dans des domaines qui pourraient sembler plutôt éloignés de l'informatique comme la biologie et la psychologie.

La partie de l'intelligence artificielle qui nous intéresse pour ce projet est celle de l'intelligence artificielle développementale. Cette branche émergente de l'intelligence artificielle prend ses sources dans la théorie constructiviste et dans les travaux du psychologue Jean Piaget.

Le constructivisme est une théorie de l'apprentissage qui repose sur l'idée que notre représentation du monde découle directement de nos expériences et observations. En interagissant avec le monde, nous nous en construisons une représentation personnelle. Cette théorie a d'ailleurs été développée par Jean Piaget dans ses travaux sur l'épistémologie génétique, théorie qui décrit l'acquisition des connaissances chez l'enfant comme étant une construction de sa propre représentation du monde au cours des premières phases de son développement.

L'intelligence artificielle développementale est donc un paradigme de conception qui consiste à considérer l'agent intelligent ou le robot comme un être à part entière qui construirait sa propre vision du monde au travers de ses expériences. Il possède un corps physique (robotique ou simulé) qui va lui permettre d'interagir avec son environnement (physique ou simulé). Ces interactions vont lui permettre d'appréhender son environnement à sa façon, sans qu'aucune connaissance préalable ne lui ait été donnée, exactement de la même façon qu'un enfant découvre le monde durant les tout premiers mois de son existence.

Les agents intelligents conçus selon cette approche vont donc découvrir leur environnement en y effectuant des expérimentations qui, combinées à leurs capacités d'apprentissage, permettent de se construire une représentation du monde. Un agent va par exemple comprendre qu'il existe dans son environnement des éléments qui l'empêchent d'effectuer certaines actions alors qu'il ne sait pas ce qu'est un obstacle ou un objet. L'un des avantages de ces agents est qu'ils sont théoriquement capables de s'adapter à tous les types d'environnements car ils n'ont besoin d'aucune information préalable sur le monde dans lequel ils évoluent. Dans le cadre du projet IDEAL Olivier Georgeon mentionne le terme "agnosticisme environnemental" (voir ci-dessous).

L'objectif de l'intelligence artificielle développementale n'est pas de construire un programme capable de résoudre un problème précis de façon efficace, contrairement à un programme qui jouerait aux échecs par exemple, mais plutôt de permettre la modélisation et l'étude des comportements. En partant du principe que l'on cherche à modéliser des comportements que l'on trouve dans la nature, donc chez les animaux ou même chez les humains, on pourrait dire que cette branche de l'intelligence artificielle permet de faire de l'ingénierie inversée sur le cerveau.

2.1.2 L'agent Ernest : un concept

C'est en se basant sur l'intelligence artificielle développementale qu'Olivier Georgeon a pu concevoir l'agent Ernest dans le cadre du projet IDEAL (Implementing DEvelopmentAI Learning). L'objectif de cet agent est de se construire une représentation de l'environnement en expérimentant et en apprenant.

Dans cette section, nous allons détailler le principe de fonctionnement de l'agent, point important pour permettre la compréhension du projet.

L'agent fonctionne sur le principe de l'interactionnisme radical (RI) [1], plutôt que d'analyser son environnement, de sélectionner la réponse appropriée et d'agir, comme c'est le cas dans les modèles classiques d'apprentissage automatique, l'agent va effectuer des interactions pour découvrir tout ce qu'il peut y faire.

Une interaction est associée à un couple (action/perception). Par exemple, l'interaction "se cogner" est associée à l'action "J'avance" et la perception "Quelque chose m'empêche d'avancer". À chaque interaction, on associe une valeur fixe que l'on appelle **valence**. Cette valeur correspond à la satisfaction "ressentie" par l'agent s'il effectue cette interaction. Elle peut être positive ou négative en fonction du ressenti que le programmeur souhaite associer à l'interaction pour l'agent ("agréable" ou désagréable"). Pour exprimer le fait qu'on réalise une interaction, on dit qu'on l'**énacte**. Une fois que l'agent aura suffisamment expérimenté, il aura tendance à éviter d'énacter des interactions marquées comme "désagréables" et privilégier les interactions marquées comme "agréables".

À chaque itération du programme, une interaction est choisie par le mécanisme de décision. Une fois cette interaction choisie, l'agent va effectuer l'action correspondante. Selon la configuration de l'environnement, une perception précise lui sera renvoyée et il saura alors quelle interaction il a effectivement réalisée. Si c'est celle qu'il a réellement voulu faire, alors le cycle de décision est considéré comme un succès, car l'agent a réussi à prédire correctement le résultat de l'expérience, et à l'inverse comme un échec si l'interaction énoncée n'est pas celle qu'il attendait. C'est à travers ces succès et ces échecs que l'agent va construire sa vision de l'environnement. Le fait d'avoir réussi à prédire l'issue de l'expérience est plus important que le résultat de l'expérience en lui-même.

À force d'effectuer des interactions, l'agent va finir par repérer des régularités, c'est-à-dire des suites d'interactions qu'il peut effectuer les unes à la suite des autres. À son lancement, l'agent ne dispose que d'un nombre restreint d'interactions ; il va être en mesure d'enregistrer les séquences d'interactions constituant

ces régularités et les considérer comme des interactions à part entière. Les valeurs permettant au mécanisme de décision de choisir une interaction seront calculées à partir des valeurs des interactions composant la nouvelle interaction.

On parle alors d'"auto-programmation", l'agent va, en créant ces nouvelles interactions à partir de séquences d'interactions plus simples, pouvoir se "programmer" et essayer de réaliser ces interactions au même titre que celles déjà présentes comme s'il "exécutait" une nouvelle séquence d'"instructions". Ce mécanisme permet une montée en abstraction des connaissances, certaines interactions encapsulent des séquences d'interactions moins complexes.

La représentation du monde que se fait l'agent Ernest est uniquement constituée des suites d'interactions qu'il peut effectuer.

2.1.3 L'agent Ernest : une implémentation

Le travail de M. Georgeon a ensuite été complété par Simon Gay au cours de sa thèse [4]. M. Gay a apporté à l'agent Ernest original de nombreuses modifications et a réalisé une implémentation open source de l'agent. Nous avons travaillé sur une version modifiée de la v7.2.¹

L'Interactionnisme Radical Parallèle

La nouvelle version de l'agent Ernest fonctionne selon le principe de l'interactionnisme radical parallèle (PRI), qui est une extension du RI. Au lieu de retourner simplement l'interaction effectivement énoncée, l'environnement retourne également un ensemble de perceptions additionnelles produites par l'interaction énoncée. Ces perceptions additionnelles sont interprétées par l'agent sous formes d'interactions dites "secondaires".

La notion d'interaction secondaire

Il existe donc des interactions secondaires qui, à l'inverse d'une interaction normale (ou primaire), ne sont plus associées à un couple (action/perception) mais à un couple (interaction/perception). Elles peuvent donc être énoncées parallèlement à une interaction primaire. Par exemple, on pourrait imaginer une interaction secondaire "Manger en avançant" qui correspondrait à l'énonciation de l'interaction primaire "Avancer", complétée par la perception "Une proie est proche de moi", et les deux interactions "Avancer" et "Manger en avançant" seraient alors énoncées simultanément.

L'agent a été doté d'un système visuel lui permettant de percevoir les couleurs autour de lui et la distance à laquelle l'agent se trouve de ce qui génère la couleur. Le système visuel renvoie donc à chaque itération un certain nombre de perceptions visuelles qui permettent l'énonciation d'interactions secondaires dites "visuelles". L'ajout de ce système visuel permet d'étendre le champ perceptif de l'agent jusqu'alors limité à ce qui entrait en contact direct avec lui.

Le parallélisme utilisé dans cette version de l'agent nous a d'ailleurs permis de manipuler des interactions secondaires qui ne sont pas directement liées aux perceptions visuelles et nous avons pu mettre en place des interactions plus complexes que celles qui étaient déjà présentes et qui nous ont été très utiles.

Les signatures d'interaction

L'agent est également capable de faire le lien entre une interaction et l'objet concerné par cette interaction. Le mécanisme de décision repose sur la construction de structures appelées signatures d'interaction. Le principe des signatures d'interaction se base sur deux constats. Premièrement, la réussite ou l'échec de l'énonciation d'une interaction est conditionnée par la présence de certaines propriétés de l'environnement, comme la présence ou l'absence d'un objet à une certaine position par rapport à l'agent. Deuxièmement, l'agent se renseigne sur son environnement via les interactions qu'il a pu énoncer, certaines interactions peuvent donc l'informer sur la présence ou l'absence de cet objet.

La signature d'une interaction est donc une structure rassemblant, pour l'interaction considérée, les interactions permettant de s'informer sur la présence de l'objet concerné. Elle permet donc de prédire, une fois apprise, le succès ou l'échec de l'interaction. Dans le code de l'agent Ernest, chaque interaction possède donc une signature représentée par un tableau de nombres flottants donnant, pour chaque interaction existante (y compris les interactions visuelles secondaires) la certitude associée à cette interaction si l'interaction propriétaire de la signature est énoncée. Les signatures sont construites au fur et à mesure de l'exécution du programme, chaque énonciation d'une interaction venant mettre à jour la signature associée.

Le programme affiche tout au long de l'exécution la représentation visuelle des signatures d'interaction.

1. liris.cnrs.fr/simon.gay/docs/Sources.zip



Figure 1 – Signature de l'interaction "Manger un Poisson" et l'agent en train d'énacter l'interaction.

Cela permet de visualiser comment l'agent définit l'objet associé à chaque interaction. Cette représentation visuelle se compose d'une série de petits carrés colorés du rouge au vert représentant la valeur de certitude associée à chaque interaction (rouge vif = -1, vert vif = 1) quand l'interaction propriétaire de la signature est énoncée. Elle comporte également deux rectangles gris sur lesquels apparaissent des couleurs. Ces couleurs correspondent à ce que l'agent associe visuellement à l'interaction, elles peuvent traduire la présence ou l'absence d'un objet. Les couleurs affichées dans le rectangle de gauche représentent des éléments directement liés à l'environnement et les couleurs affichées dans le rectangle de droite représentent l'autre agent. Par exemple, si une interaction met en jeu un élément inerte comme un mur, on verra du vert apparaître dans le rectangle de gauche et si une interaction met en jeu l'autre agent, on verra apparaître dans le rectangle de droite les couleurs liées à l'agent. La plupart du temps, les couleurs apparaissant dans les rectangles correspondent aux couleurs directement perçues par l'agent mais il existe des exceptions.

On voit sur l'exemple que l'image (cf. Figure 1) présente, sur la partie dédiée aux éléments de l'environnement, une tache mauve assez nette en son centre, de la même couleur que le poisson. On peut en conclure que l'interaction "Manger un Poisson" a été associée par l'agent à un objet mauve qui correspond donc bien au poisson.

Les représentations visuelles des signatures d'interaction nous ont été très utiles pour notre projet car elles nous ont permis, lors des tests, de s'assurer qu'en plus d'adopter le comportement voulu, l'agent avait bien associé les interactions à effectuer à l'objet attendu.

Le mécanisme de décision

La sélection de l'interaction que l'agent souhaite énoncer à chaque itération se fait via le mécanisme de décision, qui est en réalité composé de deux mécanismes distincts, le mécanisme d'apprentissage et le mécanisme d'exploitation.

Le mécanisme d'apprentissage est actif tant que les valeurs de certitude liées aux interactions sont inférieures à un certain seuil ou tant que l'utilisateur ne l'a pas désactivé. Ce mécanisme va sélectionner à chaque itération l'interaction qui a la valeur de certitude la plus faible, cela va lui permettre d'expérimenter et de comprendre dans quelles circonstances elle réussit et ainsi finir par augmenter la valeur de certitude associée.

Le mécanisme d'exploitation quant à lui est actif quand le mécanisme d'apprentissage ne l'est pas. Il va sélectionner à chaque itération l'interaction qui aura la plus grande "valeur de satisfaction globale" associée, parmi celles dont la certitude dépasse un certain seuil. Si l'interaction considérée est une interaction primaire, sa valeur de satisfaction globale est calculée en additionnant sa valence, sa valeur d'utilité ainsi que les valences et valeurs d'utilité de toutes les interactions secondaires associées si leur valeur de certitude est suffisamment élevée. La valeur d'utilité est calculée en mesurant le rapprochement des objets distants dans la mémoire spatiale en fonction du mouvement effectué si l'interaction est énoncée. Elle mesure l'utilité d'effectuer une interaction en fonction de son intérêt pour l'agent, par exemple si son énonciation permettra, dans les itérations suivantes, l'énonciation d'autres interactions qu'elle aura rendu disponibles. Si l'interaction considérée est secondaire, alors la valeur de satisfaction globale prise en compte sera celle de l'interaction primaire associée.

Dans la version du code que nous avons utilisé, le corps de l'agent et l'environnement ont été simulés à l'aide de VacuumSG, un simulateur mis au point par M. Gay.

2.2 L'apprentissage social

De la même façon qu'Olivier Georgeon a choisi de faire émerger chez l'agent Ernest des comportements simples que l'on retrouve plutôt chez des animaux primitifs, nous avons choisi de nous inspirer du monde animal pour définir d'une part ce qui constitue un comportement social et d'autre part en quoi le fait d'adopter des comportements sociaux pourrait influencer l'apprentissage.

Dans le cadre de notre projet, nous définissons un comportement social comme étant : un comportement,

donc un ensemble d'actions et réactions d'un individu dans une situation donnée, qui se déroule au sein d'un groupe social. Un groupe social étant un ensemble d'individus ayant des caractéristiques et/ou des buts communs.

Nous pouvons donc considérer que nos agents font partie du même groupe social. En effet, ils sont extrêmement similaires car conçus sur le même modèle. Ils se ressemblent physiquement, leur fonctionnement est similaire et ils sont guidés par les mêmes motivations.

Dans notre cas, un comportement social se traduirait donc par des interactions entre agents, qui seraient donc des interactions qui ne pourraient être énoncées qu'en présence d'un ou de plusieurs autres agents.

Pendant, nos agents n'ayant pas conscience d'eux-mêmes au sens où ils ne sont pas capable de comprendre leur statut d'être et n'étant pas capable d'identifier les autres agents comme membres du même groupe social, il est important de noter que ces interactions sociales n'ont de réalité que pour l'observateur du comportement.

La vie au sein d'un groupe social peut apporter de nombreux avantages aux individus, elle permet la collaboration pour l'exécution des tâches nécessaires à la survie telle que la recherche de nourriture et l'éducation des jeunes. Elle peut également permettre un apprentissage social, c'est-à-dire un apprentissage qui résulterait de l'observation des autres membres du groupe social.

Cela signifie qu'avoir un comportement social, c'est-à-dire évoluer au sein d'un groupe social et interagir avec ses membres, pourrait influencer l'apprentissage de l'environnement.

L'apprentissage social est souvent associé à des phénomènes dits de contagion [2]. Ces phénomènes peuvent être provoqués par quatre mécanismes :

- L'accentuation par l'objet
- L'accentuation locale
- La facilitation sociale
- L'imitation

L'**accentuation par l'objet** désigne l'intérêt accru que peuvent avoir un ou des individus pour un objet dans la mesure où ils ont vu un autre membre du groupe utiliser cet objet.

L'**accentuation locale** est très similaire à l'accentuation par l'objet mais se rapporte plutôt à un lieu, celui-ci est plus intéressant pour les autres dès lors qu'un membre du groupe a été observé à cet endroit.

La **facilitation sociale** est l'augmentation des chances qu'un individu effectue un comportement donné, qu'il sait déjà faire, quand il observe un ou plusieurs autres membres du groupes effectuant ce comportement.

L'**imitation** désigne l'apprentissage d'un nouveau comportement en observant un autre membre du groupe effectuant ce comportement.

3 Contribution

Notre objectif pour ce projet est de montrer qu'il est possible de modifier le code de l'agent Ernest de façon à faire émerger chez des agents un comportement social et de constater l'influence de ce dernier sur l'apprentissage.

Pour cela, nous avons imaginé différents scénarios de comportement social, que nous avons implémentés pour réaliser des expériences.

3.1 Les scénarios comportementaux

Pour décrire ces scénarios, nous nous plaçons dans l'univers de l'environnement de test où les objets et les interactions s'inspirent de concepts que l'on trouve dans la nature. Les agents sont par exemple représentés par des requins (cf. Figure 2) qui peuvent manger des poissons, ils peuvent avancer, tourner de 45 ou 90 degrés vers la droite ou la gauche et se cogner dans les murs. Il existe également dans le code la possibilité de rajouter un objet algue dans l'environnement.

Les scénarios que nous avons imaginés sont au nombre de quatre et chacun correspond à une partie de ce que nous souhaitons montrer à travers ce projet.

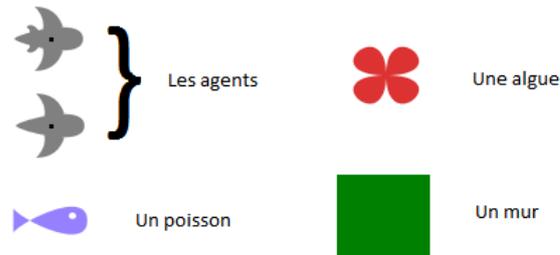


Figure 2 – Les éléments de l'environnement de test.

3.1.1 Scénario n°1 : un comportement social simple

Le premier scénario correspond à la mise en place d'un comportement social très simple d'appel/réponse : un des agents émet un signal qui correspond à un "appel". L'autre agent se rapproche alors pour répondre à cet appel pendant que l'agent appelant l'attend, immobile. Ce scénario permet de poser les bases des autres scénarios car le comportement "appel/réponse" sera également présent dans les scénarios suivants.

3.1.2 Scénario n°2 : des attitudes distinctes

Le deuxième scénario va doter l'agent d'une "personnalité" : il peut avoir un caractère agressif ou pacifique. Il est désormais possible pour un agent de toucher l'autre s'ils sont suffisamment proches l'un de l'autre. La nature de ce toucher varie en fonction de la personnalité de l'agent. S'il est pacifique, alors ce sera une expérience "agréable" pour l'autre agent et au contraire "désagréable" s'il est agressif. L'intérêt de ce scénario est d'engendrer une situation où la nature du comportement social varie en fonction de certains paramètres, permettant ainsi de savoir si des différences de comportement social provoquent des différences au niveau de l'apprentissage.

3.1.3 Scénario n°3 : l'apprentissage social

Le troisième scénario a pour but d'aborder la notion d'apprentissage social vue plus-haut via les mécanismes de contagion. Nous avons choisi de nous pencher sur le mécanisme d'accentuation par l'objet. Les agents sont en mesure d'apprendre à utiliser un nouvel objet de leur environnement, une algue rouge dans laquelle ils peuvent "jouer". De façon inconsciente, c'est-à-dire sans que cela fasse l'objet d'une action volontaire de leur part, les agents vont émettre le même signal que pour le comportement "appel/réponse" quand ils "joueront" dans l'algue, à l'image d'enfants qui communiquent inconsciemment leur joie d'utiliser un nouveau jouet. Ayant précédemment appris à réagir au signal, l'autre agent sera alors attiré par l'agent "jouant" dans l'algue, ce qui lui permettra d'être confronté à cet objet et donc d'apprendre de nouveaux comportements. L'objectif est de voir si la mise en place de ce mécanisme de contagion permet à un agent d'apprendre ce comportement de jeu dans l'algue rouge plus vite qu'un agent qui n'aurait pas été exposé à cette contagion.

3.1.4 Scénario n°4 : un comportement collaboratif

Le quatrième scénario est orienté vers la collaboration. Il existe désormais dans l'environnement un objet, l'algue rouge, qu'ils peuvent manger (comme les poissons) mais, pour ajouter la notion de collaboration, il faut être au moins deux pour y parvenir. L'intérêt de ce scénario est de savoir s'il est possible de faire comprendre à l'agent que "manger" cet objet est uniquement possible quand l'autre agent est présent et qu'il peut faire venir l'autre agent près de lui en émettant le signal. Cela traduirait donc de la possibilité de mettre en place un comportement collaboratif qui est un comportement social plutôt complexe.

3.2 Implémentation des scénarios

Nous avons ensuite pu implémenter trois des quatre scénarios décrits ci-dessus. Nous avons utilisé une base de code pré-existante en Java, la version 7.2 de l'agent Ernest, légèrement modifiée par M. Gay. Le programme permet l'exécution de l'agent Ernest dans un environnement simulé en deux dimensions. Il permet une visualisation en temps réel de l'évolution de l'agent dans son environnement ainsi que de tout les éléments calculés au cours de l'exécution, comme les signatures d'interaction, via une interface graphique.

Nous avons dû modifier le code existant de façon à rajouter tous les éléments nécessaires, à savoir les

nouvelles interactions, les actions associées ainsi que tout le code nécessaire à la détection des nouvelles perceptions. Nous avons également rendu possible la mise en place d'interactions secondaires non liées au système visuel, ce qui n'était pas prévu dans cette version du code. La mise en place des scénarios a nécessité un important travail de paramétrage, les valeurs associées aux interactions devant être finement accordées et les perceptions suffisamment précises pour permettre l'émergence du comportement attendu.

Cette partie complexe et très technique est plus précisément détaillée en Annexe A. Nous explicitons comment nous avons procédé pour gérer ces nouvelles interactions et pour développer les nouvelles mécaniques. Pour des raisons de clarté, nous avons implémenté une version du code source pour chaque scénario.

3.3 Expérimentations

Nous avons tout d'abord cherché à élaborer une démarche expérimentale adaptée à notre contexte.

Nous nous sommes donc inspirées d'un protocole classique utilisé dans les expériences sur l'apprentissage [2] et consistant à soumettre deux groupes d'individus à des expériences différentes, puis à des tests communs. Si les différences comportementales peuvent s'expliquer par les différences entre les expériences, alors il y a eu apprentissage. Les protocoles que nous avons utilisés sont des adaptations de celui-ci.

Comme chacun de nos scénarios correspondait à une partie de ce que nous souhaitions montrer au cours de ce projet, nous avons réalisé une expérimentation par scénario implémenté.

3.3.1 Scénario n°1 : un comportement social simple

La première expérience consistait à montrer que les agents sont capables d'adopter un comportement social simple, nous avons donc cherché à faire émerger le comportement "appel/réponse" décrit dans le premier scénario.

Protocole

Expérience A : les deux agents évoluent dans leur environnement, ils disposent tout deux des interactions "Appeler", "Se Rapprocher" et "Attendre".

Expérience B : les deux agents évoluent dans leur environnement, ils ne disposent pas des interactions "Appeler", "Se Rapprocher" et "Attendre".

Test Commun : on observe le comportement des agents pour savoir s'ils ont appris le comportement "appel/réponse" ainsi que les représentations visuelles des signatures d'interaction pour savoir ce qu'ils ont associé aux interactions.

Résultat

Pour interpréter les résultats de cette expérience, nous allons devoir interpréter les signatures d'interaction.

	Couleurs correspondantes dans les signatures d'interaction
Agent "éclairé" 	
Agent "normal" 	

Figure 3 – Couleurs correspondant aux différents états de l'agent.

Comme nous l'avons précisé dans la présentation du fonctionnement de l'agent, les signatures d'interaction possèdent une représentation visuelle permettant de voir ce que l'agent a associé à l'interaction. Les différents éléments de l'environnement sont représentés par une couleur.

Sur la partie droite de la signature, c'est-à-dire celle dédiée aux couleurs des éléments "agent", la couleur bleue représente l'élément agent qui appelle ou qui attend ("éclairé") (cf. Figure 3) et la couleur orange représente l'agent dans son état "normal".

On observe le comportement des agents et les signatures des interactions qui nous intéressent à partir de 2000 itérations car le comportement des agents semble s'être stabilisé.

Pour l'expérience A, on observe le comportement suivant : quand l'un des agents s'allume, son congénère se rapproche de lui de façon très directe pendant qu'il attend, c'est-à-dire qu'il reste allumé sans bouger.

On remarque même que les agents apprennent de façon complémentaire. L'un (ici, l'agent 0) apprend à appeler et attendre son congénère tandis que l'autre (ici, l'agent 1) apprend à répondre à l'appel. De façon "naturelle", c'est-à-dire sans que l'on ait cherché à provoquer ce phénomène via l'implémentation, les agents adoptent des "rôles", l'un a plus tendance à appeler/attendre et l'autre à répondre.

L'examen des signatures d'interaction confirme ce que nous déduisons de l'observation du comportement.

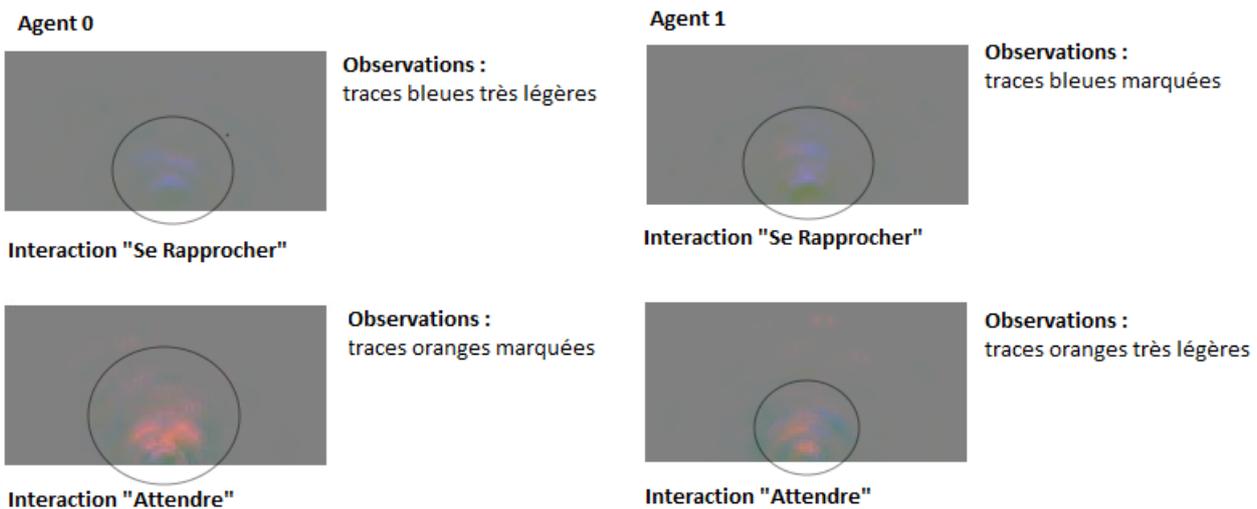


Figure 4 – Partie droite des signature des interactions "Se Rapprocher" et "Attendre" des deux agents.

Les signatures des interactions "Se Rapprocher" et "Attendre" sont très nettes (cf. Figure 4), elles présentent, sur la partie droite, des traces bleues pour les signatures de "Se Rapprocher" et des traces oranges pour les signatures de "Attendre". On peut donc en conclure deux choses.

D'une part, les agents ont associé la couleur bleue à l'interaction "Se Rapprocher", ce qui signifie qu'ils ont bien compris que l'interaction était associée à la présence de l'autre agent "éclairé". D'autre part, ils ont associé la couleur orange à l'interaction "Attendre", et donc fait le lien entre le succès de cette interaction et la présence de l'agent "normal" dans leur champ de vision. Il ont donc compris le fonctionnement de ces interactions.

Cependant, on voit que les signatures semblent aussi "complémentaires". La signature de "Se Rapprocher" pour l'agent 1 est assez nette alors que celle de l'agent 0 est moins évidente et inversement pour la signature de "Attendre".

Les signatures d'interaction se construisent petit à petit, chaque énoncé de l'interaction vient renforcer la signature de l'interaction associée. Pour obtenir une signature nette il faut avoir beaucoup énoncé l'interaction.

On peut donc déduire de cette différence de netteté une différence de comportement de la part des deux agents. L'agent 0 ayant plus tendance à énoncer "Attendre" que l'agent 1, sa propre signature de l'interaction est plus nette et vice versa.

Si on laisse tourner les simulations plus longtemps, on constate que les agents vont développer le comportement qui leur manquait et les signature des interactions vont finir par beaucoup plus se ressembler.

À l'issue de l'expérience B, les agents ne montrent aucun signe du comportement "appel/réponse". Ils évoluent dans l'environnement de façon indépendante, exactement de la même façon que lorsqu'il sont seuls. Ils n'ont donc pas développé de comportement social. Comme les interactions "Appeler", "Attendre" et "Se Rapprocher" n'existent pas, aucune signature n'est construite.

Ces résultats nous permettent donc d'affirmer que d'une part les agents sont bien capables d'apprendre à développer le comportement appel/réponse pour peu qu'on leur en donne la possibilité. D'autre part, il s'agit bien d'un comportement social puisque les interactions le composant ne sont énonçables que dans ce cas.

3.3.2 Scénario n°2 : des attitudes distinctes

L'objectif de la deuxième expérience est de montrer que l'attitude adoptée par un agent influence son apprentissage et celui de son congénère.

Protocole

Expérience A : les deux agents évoluent dans leur environnement pendant 8000 itérations, ils ont tous les deux un comportement pacifique.

Expérience B : les deux agents évoluent dans leur environnement pendant 8000 itérations, ils ont tous les deux un comportement agressif.

Expérience C : les deux agents évoluent dans leur environnement pendant 8000 itérations, l'un a un comportement agressif (agent 1) et l'autre un comportement pacifique (agent 0).

Test Commun : observation du comportement des agents entre les itérations 8001 et 9000.

Résultat

À l'issue de l'expérience A, les agents adoptent le comportement suivant : ils sont côte à côte, l'un (l'agent 1) d'eux effectue sans cesse l'interaction "Toucher" tandis que l'autre (agent 0) tourne sur lui-même ou émet alternativement les interactions "Appeler", "Toucher" ainsi que les interactions secondaires "Être Touché" associées.

À l'issue de l'expérience B, les agents adoptent le comportement suivant : ils se tournent le dos, immobiles et effectue sans cesse l'interaction "Appeler".

À l'issue de l'expérience C, les agents adoptent le comportement suivant : l'agent agressif (1) tourne sur lui-même en continu et l'agent pacifique (0) effectue sans cesse l'interaction "Appeler". Il arrive que l'agent agressif se débloque et aille manger un poisson ou toucher l'autre agent, ce qui provoque sa fuite. Les agents se déplacent alors dans l'environnement pendant quelques itérations avant de revenir à leur comportement de départ. Il leur arrive même parfois d'effectuer le comportement "appel/réponse".

L'agent Ernest est conçu de telle sorte que tout comportement est issu de l'apprentissage. On observe que les agents adoptent des comportements différents selon les expériences et le seul changement entre les expériences réside dans les attitudes attribuées à chaque agent. On constate également que l'agent 0 a des comportements différents dans les expériences A et C, alors qu'il est "pacifique" dans les deux cas. Son comportement change car l'attitude de l'agent 1 change, il est pacifique dans l'expérience A et agressif dans l'expérience B. Nous pouvons donc affirmer que la nature du comportement social a une influence sur ce qu'un agent apprend mais aussi sur ce que son congénère apprend.

3.3.3 Scénario n°3 : l'apprentissage social

Le but de la troisième expérience est de montrer qu'un comportement social permet d'augmenter la vitesse d'apprentissage de certains comportements.

Protocole

Expérience A : l'agent évolue seul dans son environnement. Il est en contact avec l'objet algue rouge.

Expérience B : les deux agents évoluent dans leur environnement, lorsqu'ils ont appris le comportement appel/réponse (2000 itérations), on les sépare en rajoutant des blocs murs dans l'environnement. L'un des agents est exposé à l'objet algue rouge pendant 500 itérations. Les deux agents sont ensuite réunis et l'agent ignorant est désormais exposé à l'algue rouge.

Test Commun : on compte le nombre d'itérations entre le moment où les agents ignorants sont exposés à l'algue rouge et le moment où ils restent jouer dedans plusieurs itérations d'affilée.

Résultat

Par souci de rigueur, nous avons effectué plusieurs fois les expériences A et B.

Expérience A (agent seul)	Expérience B (contagion)
500	2300
500	pas appris après 4500
500	100
500	400

Figure 5 – Résultats des expériences A et B en nombre d'itérations.

Comme on peut le voir dans les résultats présentés ci-dessus (cf. Figure 5), l'expérience A donne toujours le même nombre d'itérations et l'expérience B présente des résultats très variables et contradictoires.

En observant le comportement des agents, on constate que d'une part, l'agent qui a été préalablement exposé à l'algue ne s'y intéresse pas beaucoup après la fin de la période de séparation. Il ne va donc pas "jouer" dedans et par conséquent ne peut pas attirer son congénère. L'agent ignorant a plutôt, quand il finit par "jouer" dans l'algue, tendance à s'y intéresser par lui-même, comme s'il était seul avec. L'accentuation par l'objet n'a donc jamais lieu.

Ces résultats ne permettent pas d'arriver à une conclusion car d'une part ils sont trop différents et d'autre part, le comportement que nous cherchions à faire émerger pour influencer la vitesse l'apprentissage n'est pas apparu lors des tests. Nous ne pouvons donc pas dire que la vitesse d'apprentissage est influencée par le comportement social, mais nous ne pouvons pas non plus affirmer l'inverse.

Nous pensons que pour obtenir des résultats plus parlants, il pourrait être nécessaire de comprendre pourquoi l'agent "connaissant" ne s'intéresse plus à l'algue et peut-être même de réfléchir à une nouvelle façon d'implémenter le mécanisme d'accentuation par l'objet.

4 Conclusion

Au cours de ce projet, nous avons montré qu'il était possible, moyennant quelques modifications du code initial d'Ernest, de faire adopter aux agents un comportement social et que celui-ci influe sur l'apprentissage.

Nous avons proposé, implémenté et testé trois scénarios décrivant différents comportements sociaux. Les expérimentations ont permis de montrer qu'il est tout d'abord possible de doter les agents d'un comportement social très simple puis que la nature du comportement social a une influence sur l'apprentissage. Nous n'avons en revanche pas pu conclure sur l'effet du comportement social sur la vitesse d'apprentissage.

Le quatrième scénario n'a pas pu être implémenté. En cas de poursuite du projet, il nous paraît intéressant de commencer par le mettre en œuvre car cela montrerait que les agents sont capables d'adopter un comportement plutôt complexe, ouvrant ainsi la porte à des expérimentations plus ambitieuses que celles que nous avons eu l'occasion de mener.

Cependant, comme précisé en donnant la définition de comportement social, nous n'avons pas implémenté un vrai comportement social dans le sens où l'agent n'était pas lui-même conscient qu'il avait un comportement social. Pour avoir un vrai comportement social, il faudrait que les agents soient capables d'avoir conscience d'eux-mêmes et aussi capables de se rendre compte que l'agent qui interagit avec eux est un de leurs pairs, ce qui est vraiment très complexe à mettre en place chez des êtres artificiels.

De plus, les scénarios que nous avons développés et les expériences que nous avons réalisées restent très simple et sont très liés au contexte que nous avons établi. Pour aller plus loin, il faudrait imaginer et implémenter des comportements plus complexes, plus proches de la réalité.

Remerciements

Nous souhaiterions remercier M. Simon Gay qui nous a assisté tout au long du projet pour apporter à son travail les modifications nécessaires à nos expérimentations.

Références

- [1] Olivier Georgeon and David Aha. The Radical Interactionism Conceptual Commitment. *Journal of Artificial General Intelligence*, 4(2) :31–36, December 2013.
- [2] L-A. Giraldeau and F. Dubois. *Le Comportement Animal - 2ème édition*. Dunod, 2009.
- [3] H.P. Moravec. Robots and intelligence. juillet 1983.
- [4] S.Gay. *Mécanismes d'apprentissage développemental et intrinsèquement motivés en intelligence artificielle : étude des mécanismes d'intégration de l'espace environnemental*. Thèse de doctorat en informatique, Université Lyon 1, December 2014.

Annexes

Annexe A : Implémentation

Pour mettre en place les différents scénarios, nous avons donc modifié le code et mis en place de nouvelles interactions sans pour autant changer la mécanique d'apprentissage de l'agent Ernest.

Nous avons mis à disposition des agents de nouvelles interactions en les ajoutant directement dans le code, les dotant d'un nom et d'un numéro.

Il était également parfois nécessaire de définir de nouvelles actions et perceptions car elle n'étaient pas prévues dans le code.

Les actions sont très similaires, dans le code, aux interactions et ne sont identifiées que par des numéros, elles sont parfois associées à des actions concrètes quand elles induisent un changement d'état pour l'agent (comme "Avancer" ou "Tourner").

Les perceptions sont représentées par un tableau d'entiers représentant des capteurs. Il existe un capteur par type de perception, donc une case de tableau par type de perception. La valeur stockée dans la case donne la perception précise. L'ajout de nouvelles perceptions consiste donc à augmenter la taille du tableau et rajouter des tests dans la classe dédiée pour le remplir avec les bonnes valeurs.

Nous avons par ailleurs ajouté d'autres éléments permettant de mettre en place les changements qui n'étaient pas directement liés aux interactions.

Scénario n°1 : un comportement social simple

Pour le premier scénario, le comportement appel/réponse, nous avons rajouté une interaction "Appeler", associée à l'action "S'Allumer", c'est-à-dire changer de couleur pour l'agent, passant de gris foncé à bleu clair, et qui ne nécessite pas de perception particulière pour réussir, du moment que l'agent veut l'énacter, elle l'est.

Nous avons ensuite ajouté deux interactions secondaires, "Se Rapprocher" et "Attendre".

L'interaction "Se Rapprocher" est associée à l'interaction "Avancer" et à la perception complémentaire "Il y a du bleu dans mon champ de vision". Pour pouvoir réaliser cette interaction, l'agent doit énoncer l'interaction "Avancer" en ayant son congénère, éclairé en bleu, dans son champ de vision. Cette combinaison traduit un rapprochement par rapport à l'agent en train d'appeler (ou d'attendre).

L'interaction "Attendre" est associée à l'interaction primaire "Appeler" et à la perception complémentaire "Il y a du gris dans mon champ de vision", ce qui correspond à la présence de l'autre agent (non-allumé) en face de soi. Pour qu'elle soit énoncée, il faut donc que l'agent ait réalisé l'interaction "Appeler" et qu'il "voie" son congénère.

Le champ de vision mentionné ci-dessus est en réalité un champ de vision restreint, en effet, l'agent "voit" normalement tout ce qui se trouve devant lui. Pour que ses trajectoires vers l'autre soit plus directes, nous avons choisi de restreindre ce champ de vision à un tiers de sa taille originale, centré sur l'agent, pour la "vision" de l'autre. Les couleurs perçues à une itération étant stockées dans un tableau de 36 couleurs, il faut, pour que l'autre agent soit "vu", que la bonne couleur (bleu clair ou gris) soit présente dans au moins une case entre les indices 12 et 23 du tableau.

Scénario n°2 : des attitudes distinctes

Pour la mise en place du deuxième scénario, nous avons d'abord ajouté deux interactions pour implémenter le comportement "Toucher".

Nous avons ajouté une interaction "Toucher un agent" qui est associée à l'action "Toucher" et à la perception "Je suis très proche de l'autre agent". La proximité étant calculée directement à l'aide des coordonnées des agents dans l'espace.

Ensuite, nous avons créé une interaction "Toucher le vide", associée à l'action "Toucher" et à la perception "Je ne suis pas à proximité de l'autre agent". Cette interaction est "alternative" à l'interaction "Toucher un agent". Cela signifie que l'échec de l'énaction d'une des deux interactions entraîne systématiquement l'énaction de l'autre. Par exemple, si l'agent tente de réaliser l'interaction "Toucher un agent" mais qu'aucun autre agent n'est à sa portée, c'est l'interaction "Toucher le vide" qui sera énoncée.

Nous avons dû créer cette interaction alternative car l'agent doit obligatoirement énoncer une interaction à chaque itération, il était donc obligatoire de fournir une autre interaction qui pourrait servir de substitut au

cas où l'énaction de l'interaction "Toucher un agent" échouerait.

Nous avons ensuite ajouté une interaction secondaire par interaction primaire existante et dont la perception complémentaire est "Je suis très proche de l'autre agent et il est en train de toucher". Ces interactions secondaires permettent à l'agent de savoir qu'il a été touché par l'autre et comme cela peut survenir à n'importe quel moment, il était nécessaire d'associer une interaction secondaire à chaque interaction primaire.

Pour mettre en place les "caractères" nous avons donné à l'agent un tableau de nombres flottants qui possède autant de cases qu'il existe d'interactions dans le code. Chaque case correspond donc à une interaction et la valeur stockée dans la case va venir moduler la valence associée à l'interaction. À chaque fois que la valence est utilisée pour un calcul, elle est multipliée par la valeur correspondante du tableau. Ce mécanisme nous permet ainsi une grande souplesse d'implémentation des traits de caractère, sans modifier de façon importante le code existant.

Pour mettre en place nos "personnalités", nous avons choisi de ne pas implémenter le caractère au sein de l'agent, mais plutôt de contrôler directement la réaction son congénère à son contact en modulant le ressenti (la valence) des interactions secondaires correspondant à la perception du toucher.

Pour mettre en place le caractère "pacifique", nous avons donc fait en sorte que le fait d'être touché par l'agent pacifique soit "agréable". Il fallait que les valeurs de satisfaction des interactions secondaires "être touché" soient positives et comme les valences de ces interactions sont positives par défaut, nous avons choisi de mettre la valeur 1 pour ces cases dans le tableau de l'autre agent. Pour le caractère agressif, nous avons mis ces valeurs à -1, car il fallait que l'expérience soit "désagréable" pour l'autre agent.

Dans les deux cas, l'agent apprécie de toucher l'autre et les interactions restantes ne sont pas affectées par la nature du caractère, les valeurs des autres cases du tableau sont donc fixées à 1.

Ici, le caractère de l'agent est donc déterminé par les valeurs stockées dans le tableau de l'autre agent mais cela est très lié à la nature de la personnalité que l'on a souhaité attribuer aux agents. On peut très bien imaginer que la mise en place d'autres attitudes n'entraînera pas ce besoin de modification croisée et que l'on pourra se contenter de modifier le tableau de l'agent concerné.

Scénario n°3 : l'apprentissage social

La mise en place du dernier scénario a simplement nécessité l'ajout de deux interactions. L'ajout d'une interaction "Jouer dans le vide" qui correspond à l'action "Jouer" et à la perception "Je ne suis pas près d'une algue" et l'ajout d'une interaction "Jouer avec une algue" qui est associée à l'action "Jouer" et à la perception "Je suis près d'une algue". Ces deux interactions sont, comme "Toucher le vide" et "Toucher un agent", alternatives.

Si l'interaction "Jouer avec une algue" est effectivement énoncée, c'est-à-dire si l'agent cherche à effectuer l'interaction "Jouer avec une algue" et qu'il est près d'une algue, alors il s'allume, (sa couleur passe de gris à bleu clair) comme pour les interactions "Appeler" et "Attendre".

Ces modifications n'ont pas été sans conséquences pour le code. En effet, la mise en place du scénario numéro 2 a occasionné un ralentissement considérable dans l'exécution car nous avons ajouté beaucoup d'interactions, augmentant du même coup le nombre de calculs à effectuer à chaque itération.