République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

MEMOIRE

Présenté par

Miroud Aymen

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Elicitation des préférences dans un domaine de négociation multi-issues non linéaire.

Soutenu le : / / 2022

Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mr. Benmachiche	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mr. Betouil A.A**	**MCB**	**Chadli Bendjedid El-Tarf**
Examinateur	Mr. Bentrad S	MCB	Chadli Bendjedid El-Tarf

Année Universitaire : 2021/2022

Remerciements

Tout d'abord je tiens à remercier ALLAH qui m'a donné la force, l'intelligence et la patience d'accomplir ce modeste travail.

Mes remerciements et ma gratitude s'adressent à mon encadreur ***Dr Betouil Ali Abdelatif*** *pour sa disponibilité et l'intérêt qu'il a porté à mon travail tout au long de semestre.*

Un grand merci ainsi que tous les enseignants de département Mathématique et informatique.

Enfin, je remercier mes chers parents, ma famille et mes amis qui m'ont encouragée tout au long de mes études.

Leur soutien et leur aide sont pour beaucoup dans l'accomplissent de ce travail

Dédicace

Du profond dans mon cœur, je dédie ce travail à tous ce qui est chères.

A mes chers parents,A ma tante

aucun dédicace ne serait exprimer mon respect, mon amour éternel et ma considération pour les sacrifies que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout soutient, la motivation et l'amour que vous portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Que ce modeste travail soit l'exaucement de vos vœux tout formulés, le fruit de vos innombrablessacrifies. Puisse Dieu le très Haut, vous accorder santé, bonheur et longue vie.

A ma belle-sœur Meriem, et mes frères Chihab et Amir

C'est un moment de plaisir de dédier ce travail, en signe d'amour, de reconnaissance et de gratitude pour le dévouement dont vous avez fait toujours preuve à mon égard.

A ma fiancée Abir

merci énormément pour ton soutient plus que précieux, merci pour ton grand cœur

Résumé :

La négociation est un processus par lequel toutes les parties peuvent parvenir à une solution. Dans le cadre de ce travail, nous abordons les mécanismes de négociation bilatérale multi issues dans un domaine non linaire à fin d'obtenir une bonne solution de compromis.

Le but est de proposer une approche qui permet de modéliser les préférences d'adversaire en basant sur les négociations passées.

Dans ce travail, nous développons un agent basé sur l'apprentissage par renforcement (AR) et implémenté via l'algorithme SARSA pour prédire les intentions d'autres agents et les actions futures possibles. Afin de maximiser le rôle de notre agent et en fonction de la décision de sélectionner la meilleure offre parmi plusieurs offres multi-issue non linéaires (y compris les stratégies acceptation / refus, concessions et proposition).

Nous montrons que notre agent est capable d'apprendre dans un environnement non linéaire

multi-issue et obtient une valeur d'utilité meilleure que les autres agents disponibles dans la plateforme de simulation des négociations automatiques appelée GENIUS.

Les mots clés : négociation automatique, agent, apprentissage par renforcement, GENIUS, multi issues non-linéaire

Abstract:

Negotiation is a process by which all parties can reach a solution. In this work, we approach the mechanisms of bilateral multi-issue negotiations in a non-linear field in order to obtain a good compromise solution.

The aim is to propose an approach that allows modeling of adversary preferences based on past negotiations.

In this work, we develop an agent based on reinforcement learning (RL) and implemented via the SARSA algorithm to predict the intentions of other agents and possible future actions. In order to maximize the utility of our agent and depending on the decision to select the best offer among several non-linear multi-issue offers (including acceptance / refusal, concessions and proposal strategies).

We show that our agent is able to learn in a non-linear multi-issue environment and obtains a value in use better than the other agents available in the automatic trading simulation platform called GENIUS.

The key words: automatic negotiation, agent, reinforcement learning, GENIUS, non-linear multiissue

ملخص:

التفاوض هو عملية يمكن من خلالها لجميع الأطراف التوصل إلى حل.كجزء من هذا العمل ، نتناول آليات التفاوض الثنائي متعدد القضايا فيمجال غير خطي من أجل الحصول على حل وسط جيد.الهدف هو اقتراح نهج يسمح بنمذجة تفضيلات الخصمعلى أساس المفاوضات السابقة .

في هذا العمل ، نقوم بتطوير وكيل يعتمد على التعلم المعزز (AR) ويتم تنفيذها عبر خوارزمية SARSA للتنبؤ بنوايا الوكلاء الآخرين والإجراءات المستقبلية المستطاع. من أجل تعظيم دور وكيلنا واعتمادًا على قرار اختيارأفضل عرض من بين العديد من العروض متعددة المشكلات غير الخطية (بما في ذلك القبول /الرفض والتنازلات والاقتراح).

نظهر أن وكيلنا قادر على التعلم في بيئة غير خطيةمتعدد ةالقضايا ويحصل على قيمة فـي الاستخدام أفضل من العوامل الأخرى المتاحة في منصة المحاكاة للمفاوضات التلقائية تسمى GENIUS.

الكلمات المفتاحية: التفاوض التلقائي ، الوكيل ، التعلم المعزز ، GENIUS ، القضايا المتعددة غير الخطية.

Table des matières

Liste des figures

Liste des tableaux

***Glossaire**:*

α ……Coefficient d'apprentissage.

ε ……..Taux d'exploration.

γ ……Coefficient d'actualisation.

P ……Probabilité de transition.

R ……Fonction de récompense.

r ……Récompense scalaire.

t……Temps Période d'échantillonnage.

T……Deadline.

a……Action.

A……Ensemble d'action.

s ……Etat courant.

S…… Ensemble des états.

Q …… Fonction d'utilité.

$U_A(w_{B\to A}^t)$……L'utilité de l'agent « A » qui lui gagner d'après la proposition de « B ».

$U_A(w_{A\to B}^t)$……L'utilité de l'agent « A » qui lui gagner d'après leur proposition chez l'agent « B ».

β……Une valeur réservé de l'utilité minimale.

Introduction Générale

L'intelligence Artificielle (IA) est une branche des sciences de l'informatique qui s'intéresse à la reproduction de certains aspects de l'intelligence humaine dans le but de développer des algorithmes, des méthodologies et des systèmes destinés à la résolution de problèmes complexes. L'approche classique de l'IA aborde cette question d'une manière centralisée, ainsi un seul processus monopolise la gestion des connaissances, la manipulation des ressources et la réalisation des tâches. Cette approche centralisée a été remise en question vu les inconvénients qu'elle présente, en particulier la nécessité d'intégrer au sein d'une même entité des connaissances et des compétences diverses et nombreuses, qui dans la réalité appartiendraient à plusieurs individus collaborant ensemble pour réaliser un but commun. De ce constat est né le besoin de passer d'une vision purement individuelle de la résolution d'un problème à une vision collective qui nécessite la distribution de l'intelligence sur plusieurs entités. C'est ainsi qu'est apparue l'Intelligence Artificielle Distribuée (IAD). L'IAD a elle-même introduit un nouveau paradigme, il s'agit du paradigme multi-agents. Dans ce dernier, on s'intéresse à la manière de répartir un problème sur un certain nombre d'entités autonomes appelées agents, capables d'agir dans un environnement déterminé, de communiquer, de coopérer et de résoudre les éventuels conflits dans le but de réaliser un objectif commun.

Les Systèmes Multi-Agents (SMA) ont, depuis, occupé une place importante au cœur des nouvelles technologies, et se sont positionnés au carrefour de plusieurs autres disciplines connexes où ils puisent leurs sources d'inspiration comme la sociologie, la philosophie, l'éthologie, les réseaux, etc. Ils s'avèrent d'une grande efficacité dans des champs d'application variés où les approches de développement classiques restent limitées. Les SMA sont particulièrement adaptés aux systèmes complexes, ouverts, distribués, asynchrones et dynamiques. Ils sont utilisés dans plusieurs applications comme la production industrielle, la supervision de processus, la robotique, le diagnostic, le trafic aérien, le commerce électronique, la recherche d'informations, etc.[1]

Notre travail se concentre sur l'intelligence artificielle, en particulier l'intelligence artificielle distribuée et les systèmes multi-agents. Lorsque plusieurs agents interagissent, des conflits peuvent survenir, ce qui nécessite l'utilisation de mécanismes de résolution des

conflits. Ces mécanismes comprennent la coordination, le système de vote et la négociation..

La négociation est un moyen d'amener les agents à résoudre un état de conflit ou de compétition à travers la communication. Cette approche permet à chaque agent de faire valoir son point de vue et ainsi de satisfaire au mieux les intérêts respectifs de chacun. La négociation permet, elle, une prise en compte des objectifs personnels pour atteindre un accord avec le meilleur compromis. Il existe deux formes de négociation agent : compétitive ou coopérative. Pour la première, les agents vont négocier en partageant des informations affin de tenter des choix de groupe pour chaque action conflictuelle. Ils évaluent alors l'utilité des propositions en considérant jusqu'où ils sont prêts à "payer" pour débloquer la situation par rapport au "coût" de la proposition. Un ensemble de négociation correspond alors à tous les accords ayant une utilité positive pour chaque agent et représentant autant de solutions possibles au problème. Cette forme de négociation peut se baser sur la théorie des jeux , les heuristiques , la vente aux enchères ou encore sur l'argumentation. La négociation coopérative implique, elle, une collaboration des agents impliqués. [21]

Dans ce cadre, chaque agent apporte des offres directement au prix le plus élevé pour eux. Ils considèrent même la possibilité d'avoir une utilité négative dans une certaine mesure afin de débloquer le problème. Divers protocoles suivent cette approche, comme par exemple une proposition de protocole liée à une méthode d'ordonnancement de lignes de production . Ce type de négociation reste principalement appliqué avec le protocole d'interaction de "réseau contractuel" .

La négociation qui nous allons traiter dans cette mémoire est une négociation compétitive, car nous sommes intéressés à maximiser l'utilité de notre agent négociateur. . La recherche sur la négociation automatique peut être décomposée en trois larges thèmes :

-**Protocole de négociation :** un ensemble de règles pour contrôler l'interaction.

-**L'objectif de la négociation :** le périmètre standard de la solution doit être atteint. Dans un cas extrême, l'objectif ne peut contenir qu'une seule issue (par ex. Prix), mais il peut également contenir des multi issues (avec le prix, Qualité, délai, pénalités, conditions générales, etc.).

-**Les modèles décisionnels des agents :** méthode permettant à un agent d'atteindre ses objectifs de négociation tout en suivant les règles de négociation. Complexité L'accord

utilisé, la nature des objectifs de négociation et la gamme d'opérations qui peuvent être appliquées.

Dans ce mémoire nous intéressons au domaine des modèles décisionnels des agents, notre but est de fournir une stratégie d'apprentissage qui permet à l'agent d'anticiper l'intention et les futures actions possibles de l'autre agent adversaire au cours d'une négociation multi issues et non linéaire, où l'agent cherche de maximiser son utilité local et gagner le tour de négociation. Les modèles utilisés par les agents sont non linéaires, car les agents doivent établir leur propre représentation à travers de multiples interactions avec l'environnement.

Ainsi, de créer des stratégies d'acceptation ; c'est la condition choisi par l'agent selon leur veut de maximiser leur utilité, à côté d'une stratégie de concession qui présente un cas particulier le temps qu'il renonce à certain condamnations à son profit, et une stratégie de proposition faite où l'agent refus un offre d'adversaire et il lui proposée une contre-offre pour augmenter leur utilité.

Notre défi dans ce travail est d'effectuer une elicitations des préférences dans un domaine de négociation multi issues no linéaire selon une modélisation et un algorithme basé à l'apprentissage par renforcement.

Dans ce travail nous développerons un agent capable de gagner une négociation avec la maximum valeur d'utilité à l'utilisation des fonctions de l'apprentissage par renforcement et l'algorithme SARSA.

Pour ce faire ; nous organisons notre mémoire en quatre chapitres :

Dans le premier chapitre : nous présentons la notion de base des systèmes multi agent et ça composant de base, ainsi qu'une vue presque détaillée sur la notion d'agent et leur caractéristiques.

Le deuxième chapitre est composé de deux parties, la première partie présente la négociation, ses composants de base, les modèles, les protocoles de négociation et une validation des protocoles, et les stratégies de négociation. Dans La deuxième nous fournissons une définition sur la négociation multi issues et nous prend un modèle d'une négociation multi issues.

Dans le troisième chapitre : nous présentons une brève description de notre futur agent a modélisé ; nous appuyons sur l'apprentissage par renforcement ; ensuite nous confirmons le choix d'algorithme SARSA et commencer à la conception de notre agent

Dans le quatrième chapitre : nous montrons la partie expérimentale de notre travail et discuter les résultats.

Chapitre 01:

Systèmes multi-agents

Introduction :

Les systèmes multi-agents (SMA) sont une branche de l'intelligence artificielle distribué(IAD) ; alors on peut dire que les SMA sont un domaine multidisciplinaire, on fait, les concepts de ce domaine sont d'origine diverses.

SMA est un nouveau paradigme de programmation qui a occupé de plus en plus une place importante parmi les technologies de développement systèmes complexes et distribués. Cette importance peut se refléter sur l'évolution du marché mondial des agents logiciels. Dans ce chapitre on présentant les concepts généraux des systèmes multi agents.

1-Définitions et spécifications

Les systèmes multi-agents sont issus de l'intelligence artificielle distribuée. Ferber indique qu'ils émergent d'une remise en question de l'informatique séquentielle en considérant que les activités sont résolues par interaction et coopération d'entités autonomes appelées agents [2]

Différentes définitions de l'entité agent existent mais trois principales se démarquent et font encore aujourd'hui office de références.

Selon Ferber, un agent est défini comme "*une entité physique ou virtuelle évoluant dans un environnement dont il n'a qu'une représentation partielle et sur lequel il peut agir*"[3] . Cet agent peut "*communiquer avec d'autres agents et est doté d'un comportement autonome*". A travers cette définition nous relevons quelques points clefs :

l'agent se caractérise par sa capacité à communiquer avec l'environnement et les autres entités agents. Par ailleurs, l'accent est mis sur la notion d'autonomie, c'est à dire la capacité des agents à évoluer sans intervention directe de l'utilisateur.

Nous pouvons retrouver ces éléments dans la définition de Demazeau et Costa établissant un agent comme "*une entité réelle ou virtuelle dont le comportement est autonome, évoluant dans un environnement qu'il est capable de percevoir et sur lequel il est capable d'agir, et d'interagir avec les autres agents*" [4]. Ici la notion d'interaction est soulevée. Elle regroupe l'idée de communication, d'action et de perception des éléments entourant l'agent. Nous retrouvons également l'idée d'autonomie.

Enfin, selon Wooldridge un agent est "*un système informatique capable d'agir de manière autonome et flexible dans un environnement*" [5].

Il précise le terme de flexibilité avec trois éléments : la réactivité, la pro-activité et les capacités sociales. Il définit ainsi "*un système réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent*", "*un système pro-actif génère et satisfait des buts. Son comportement n'est donc pas uniquement dirigé par des événements*" et "*un système social est capable d'interagir ou coopérer avec d'autres systèmes*". Nous retrouvons toujours cette notion d'autonomie à laquelle Wooldridge présente ainsi trois caractéristiques supplémentaires aujourd'hui essentielles pour définir un agent. L'architecture générale d'un agent. [Figure 1] illustre l'interaction avec l'environnement et les autres entités agents le composant par le biais d'acteurs et de capteurs. Les choix d'action sont dictés par une notion de réaction et/ou décision de l'entité agent. Cette notion est induite par les connaissances et compétences de l'entité agent.

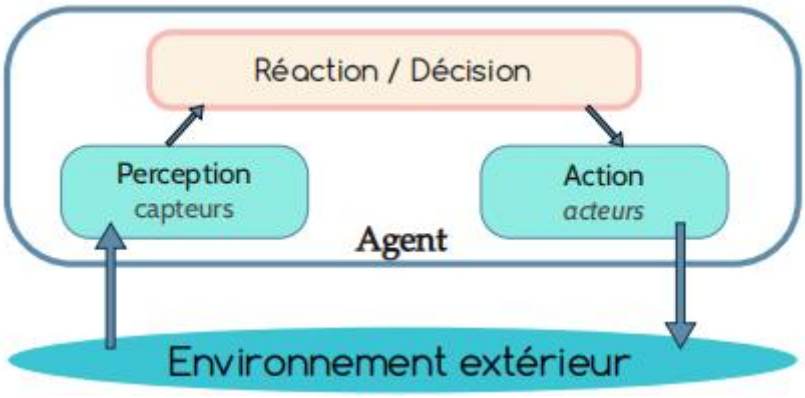


Figure 1 – Architecture générale d'un agent. [6]

Nous souhaitons expérimenter les algorithmes agents au cœur de systèmes embarqués pour permettre leur optimisation. Nous sommes particulièrement intéressés par la capacité d'interaction des agents. Leurs compétences et connaissances sont mises en avant pour faire ressortir leur capacité à rendre des services. Cette caractéristique nous conduit à la définition d'un agent [Définition 10].

Définition #10 :

Un **agent** est une entité physique ou virtuelle autonome capable d'interagir avec son environnement dont il possède une représentation partielle. Il peut également interagir avec les autres agents partageant cet environnement.

Un agent possède des compétences et connaissances propres. Il peut les combiner pour ofrir des services.

Par exemple dans le cadre d'un vol d'oiseau, l'agent oiseau peut interagir avec les autres oiseaux pour décider des directions de vol. Il prend également connaissance de son environnement pour repérer les obstacles. En regroupant ces agents en systèmes multi-agents, nous obtenons des ensembles de nature distribuée capables de générer de multiples interactions. Cette capacité apporte aux systèmes multi-agents la possibilité d'évoluer en s'adaptant à leur environnement. Leur faculté à prendre des décisions et à agir sans intervention d'un tiers participe à leur autonomie [6].

Les SMA se concentrent sur une vision décentralisée de la formalisation et résolution des problèmes [Défénition 11]. Par exemple, dans le cadre de simulations, des voitures sur une portion de route peuvent représenter un SMA où chaque voiture est une entité agent.

Définition #11 :

Un **système multi-agents** est un système constitué de plusieurs entités agents situées dans un même environnement et interagissant pour répondre à des requêtes ou objectifs précis.

1-1-Différentes catégories d'agents

Les agents peuvent être classifiés dans trois grandes catégories : les agents dit réactifs, cognitifs ou hybrides [3].

• Agent **réactif**.

Comme son nom l'indique, il réagit aux stimuli perçus dans l'environnement. Ses actions sont déterminées par des règles préétablies et les conditions perçues. Un agent réactif peut intégrer un état interne qui sera partie intégrante des conditions à analyser dans le choix de l'action à effectuer. Les règles régissant les actions sont établies selon le service que doit accomplir l'agent. Il peut s'agir d'un besoin interne (se nourrir pour un animal, maintenir son

niveau énergétique pour une machine), ou d'une tâche spécifique à accomplir. Les agents réactifs ne possèdent pas de vision à long terme de leur environnement. Ils ne prévoient donc pas leurs actions futures selon une planification des changements de l'environnement. En contrepartie, ils sont robustes et fiables car ils proposent des comportements redondants basés sur des réactions à des effets de seuils. Leur empreinte mémoire est moindre et ils interagissent rapidement avec leur environnement. Leur force vient des comportements issus de leurs interactions et permettant leur adaptation avec le SMA et leur environnement.

• Agent **cognitif**.

Il possède des raisonnements internes basés sur des buts à réaliser et une représentation de son environnement et des autres agents qui l'entourent. L'agent cognitif détient une certaine mémoire des événements passés. Il est ainsi capable de raisonner sur ce qu'il perçoit de son environnement, sur les impacts qu'auraient ses actions et sur la satisfaction qu'il tirerait de ces impacts. Il décide en fonction de ces raisonnements quelle action appliquer. Il peut ainsi mémoriser des situations et des réactions de l'environnement à ses actions. Cette faculté lui permet de planifier son comportement en prévoyant sur le long terme les fluctuations de l'environnement et les actions les plus appropriées. Les agents cognitifs sont plus gourmands en termes d'empreinte mémoire ou de temps de calcul. En effet, chaque décision passe d'abord par une étape d'analyse augmentant leur temps de réponse en comparaison des agents réactifs. Par ailleurs, il est difficile de déterminer comment l'agent évoluera à partir de sa base de règles puisqu'il apprend de son environnement. Cette spécificité peut créer des biais de respectabilité. En contrepartie, leurs capacités de planification, d'analyse et d'apprentissage permettent à certains agents d'optimiser leur comportement par rapport à leur environnement.

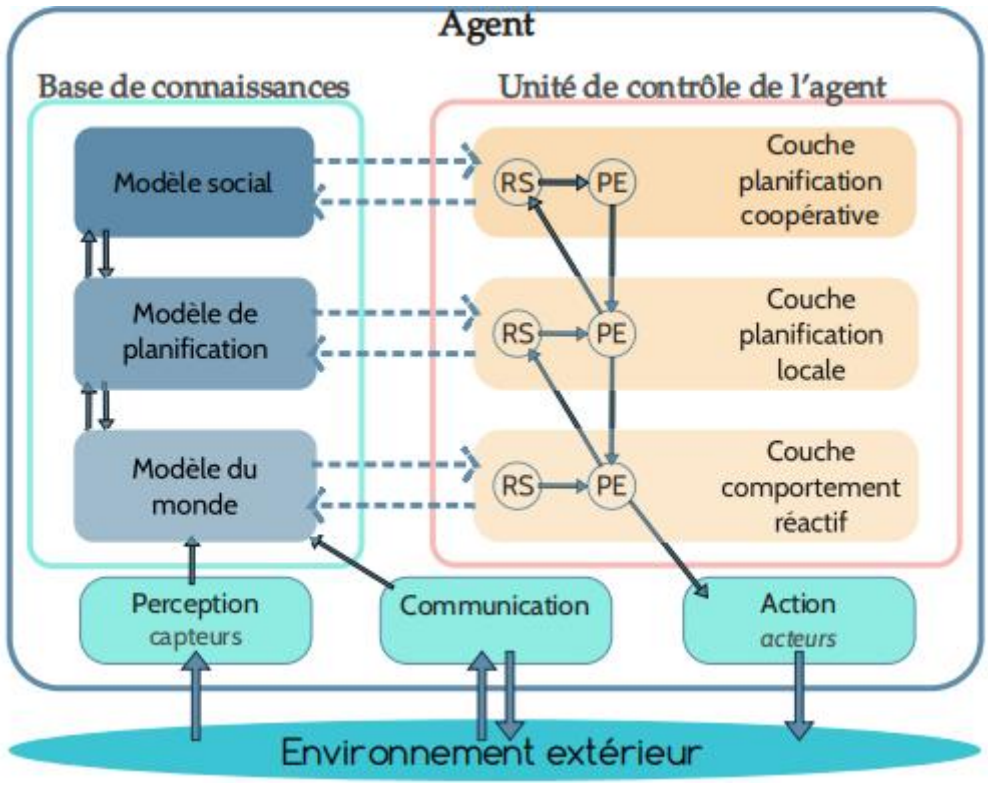


Figure 2 – Architecture d'un agent hybride ilustrée par l'exemple InterRaP.[7]

• Agent **hybride**.

Un exemple d'architecture d'agent hybride est illustré en Figure 2 [7]. Dans cet exemple, les agents hybrides sont construits sur un modèle de couches pour assurer les deux aspects vus précédemment : réactif et cognitif. La couche la plus bas niveau correspond à une approche réactive. Elle effectue sa Reconnaissance de Situation (RS) directement à partir des données brutes de la base de connaissances fournies par les capteurs et la communication avec les autres agents. Elle est capable de prendre des décisions et d'initier des actions (Planification et Exécution, PE). Cette étape et les actions en découlant pourront être mises à jour par les couches supérieures.
La couche intermédiaire travaille avec une vision plus globale de l'environnement. Sa reconnaissance de situation reste à partir de données locales. La dernière couche représente la partie cognitive. Elle ajoute les aspects sociaux aux connaissances locales. L'agent peut alors se représenter les autres agents partageant son environnement (leurs buts et connaissances, etc.). La reconnaissance de situation permet de prendre en compte le long terme dans le choix d'exécution.

Ces différentes catégories d'agents ont longtemps défini la nature de ces derniers. L'évolution des différents modèles proposant des représentations plus variées des entités agents ont amené une convergence des caractéristiques vers une notion de comportement. Cette notion permet de concentrer la description de l'entité sur ses actes sans l'astreindre à une catégorie fixe.

1.2. Modèles agents

Un modèle correspond à la formalisation d'un SMA. Il détermine les agents qui le composeront, leurs actions, leurs interactions, leur forme de réflexion, leur capacité de décision, etc. Nous citons ici une liste non exhaustive de modèles tout en souhaitant rester représentatifs de la diversité des approches.

Certains modèles s'appuient sur l'approche objet, comme la méthode MaSE (*Multiagent Systems Engineering*) [8]. Ce modèle délimite son fonctionnement à des SMA fermés, dont l'environnement n'évolue pas. Ces environnements doivent également être statiques, à savoir composés d'agents dont l'état ne change pas durant l'exécution. Par ailleurs, le modèle considère uniquement les interactions d'un agent vers un autre agent en opposition

aux interactions à multiples destinataires. MaSE se découpe en deux parties : d'abord une phase d'analyse des spécifications du SMA, puis une phase de conception guidée. La phase d'analyse comprend la spécification des objectifs du système à partir d'un cahier des charges, la formalisation des cas d'utilisation à travers des conversations inter-agents et enfin la transformation de ces cas d'utilisation en rôles en considérant les objectifs définis. La phase de conception comporte la réalisation de classes d'agents à partir des rôles définis, la réalisation des conversations formalisées, l'assemblage des différentes classes en un SMA et enfin le déploiement de ce dernier. Cette forme de modèle développe peu le concept social de l'agent du fait de la limitation des conversations à des cas d'utilisation pré-établis.

D'autres modèles choisissent plutôt de s'appuyer sur une architecture agent BDI, basée sur la psychologie du raisonnement et largement validée par la communauté de recherche [9]. Ce modèle repose sur trois ensembles inter-connectés : les croyances, les désirs et les engagements. La notion de croyance traduit la base de connaissances de l'agent et lui donne une certaine vision du monde auquel il appartient. Cette base de connaissances est mise à jour par l'agent grâce à la perception de son environnement et des interactions avec les autres entités agents. Ces croyances ne sont pas forcément vraies, elles peuvent être faussées par les perceptions de l'agent, ou simplement être tronquées dès le départ. Les désirs d'un agent sont générés à partir de ses croyances. Ils reflètent à la
fois les opportunités de l'agent à un instant donné et ses objectifs à long terme. Ils nécessitent une ou plusieurs actions pour être remplis. L'agent va parfois décider de passer à l'acte. Il est alors question d'engagement pour signifier que le désir a été retenu. Avec ce modèle, le comportement de l'agent va évoluer en fonction de ses croyances mais aussi avec les expériences accumulées suite aux différents engagements.

Parmi les méthodes utilisant cette architecture d'agents, nous retrouvons par exemple la méthode AAII (Australien Artificiel Intelligence Institute Methodology) développée pour la gestion du trafic aérien [10]. Nous pouvons également citer la méthode DESIRE (*DEsign and Specfication of Interacting REasoning framework*) issue de l'ingénierie des connaissances [11] [12].
Une autre méthode reprenant l'héritage des approches d'ingénieries des deux modèles précédemment cités est Gaia [5]. Ce modèle, se voulant plus générique, considère les agents comme des systèmes informatiques approximatifs reliés à l'utilisation d'une

ressource. L'objectif du SMA développé par Gaia est d'optimiser la qualité de certaines mesures générales même si la solution émergente n'est pas optimale d'un point de vue matériel. Par conséquent, cette méthode se décrit elle-même comme non adaptée dans le cadre de conflits de ressources. Tout comme la méthode MaSE, Gaia considère uniquement des SMA statiques. Les services et la base de connaissances sont eux spécifiés à partir des rôles et des interactions déterminés durant la phase d'analyse. Cependant elle ne prend pas en compte les systèmes devant gérer des conflits.

L'architecture agent AGR (Agent Groupe Rôle) définit les rôles joués par l'agent au sein de groupes, avec une vision centrée organisation. La méthode Ala adin propose un cadre solide de développement de systèmes multi-agents à partir de cette forme d'architecture [13].

Les modèles pour les systèmes multi-agents peuvent se baser sur l'organisation des agents selon leur rôle (AAII, Aalaadin), leurs buts (MaSE) ou leurs tâches (DESIRE, Gaia). Pour nos travaux, les agents pourront être répartis sur plusieurs niveaux de l'architecture matérielle avec différentes contraintes. Il est donc important de rester indépendants des spécifications de l'agent lors de la formalisation du SMA pour garantir une meilleure adaptation à notre contexte d'application. Pour cela, nous nous intéressons aux algorithmes agents liés à leurs interactions et partons donc d'un modèle agent centré sur ce point. Deux modèles correspondent à ces attentes : DIAMOND (*Decentralized Iterative Multiagent Open Networks Design*) [14] et IODA (*Interaction-Oriented Design of Agent simulations*) [15].

1.2.1DIAMOND

DIAMOND utilise le modèle AEIO (Agent Environnement Interaction Organisation) privilégiant une description explicite des interactions et de l'environnement de l'agent [16]. DIAMOND est une méthode de co-design ; elle offre une formalisation de conception hybride traitant conjointement le logiciel et le matériel. Cette pratique implique que toutes les étapes de la méthode sont pensées pour une concordance optimale entre les deux parties. La particularité de DIAMOND est d'être dédiée à la spécification de logiciels multi-agents [17].

Le cycle de vie de DIAMOND est un modèle en spirale composée de plusieurs cycles, chacun découpé en quatre phases : la détermination des besoins du cycle à venir, l'analyse des risques, le développement et la validation de la solution retenue. Cette dernière phase implique la revue des résultats obtenus et la planification du cycle suivant. Ce modèle offre donc la possibilité de remettre en cause les besoins et la finalité du produit à chaque cycle. Au sein de ce cycle, une section de la méthode est dédiée à la réalisation du modèle multi-agents considéré comme ensemble logiciel.

Cette partie comprend l'analyse des besoins logiciels, la conception du logiciel et l'expérimentation du logiciel. Elle reste cependant très liée au développement du reste du système pour assurer une certaine symbiose. Ainsi, les choix de développement du système embarqué permettent de combler des besoins du SMA et réciproquement.
La phase de conception générique développe la formalisation du SMA. DIAMOND y spécifie d'abord son contexte d'utilisation. Ensuite, la méthode propose la construction des tâches applicatives agent et la construction des modules de communication et des structures d'organisation. Enfin, DIAMOND détaille la construction du contrôle de l'agent. L'ensemble de cette partie s'appuie sur la documentation générée lors des précédentes phases de définition des besoins et d'analyse de DIAMOND. Notamment l'étape d'analyse étudie l'environnement dans lequel les entités évoluent, puis à l'axe agent, aux interactions, et enfin l'organisation de l'ensemble du système, selon la décomposition AEIO.

Dans la phase de conception générique, l'itération multi-agents passe également par ces quatre axes d'étude à travers les parties suivantes :
- le contexte d'utilisation spécifie des situations pour définir les limites du système multi-agents et de son environnement ;
- la construction des tâches applicatives agent détaille l'agent de façon interne et individuelle ;
- la construction des modules de communication et des structures d'organisation représente une phase sociale spécifiant les interactions et l'organisation des agents ;
- enfin, la construction du contrôle de l'agent correspond à une phase d'intégration des influences sociales au cœur des agents.

DIAMOND se concentre ensuite sur la réalisation complète d'un système embarqué adapté aux besoins physiques du système multi-agents spécifié. Pour notre part, nous souhaitons

partir d'un SE existant et y associer un SMA pour proposer une nouvelle application d'optimisation au sein de ce système embarqué sans en modifier les composants.
Dans le cadre de travaux joignant les systèmes multi-agents aux systèmes embarqués, notre démarche représente donc plutôt une certaine complémentarité avec l'approche DIAMOND.

1.2.2 IODA

IODA est un modèle multi-agents basant sa réalisation sur les interactions entre agents et leur spécification [18]. L'avantage est de proposer une formalisation générique des communications, indépendante du contexte d'utilisation. Le modèle IODA part des Interactions et de l'Environnement pour ensuite s'intéresser à l'Organisation et enfin à l'Agent. Les contraintes de l'environnement matériel peuvent donc être prises en compte très tôt dans la procédure. L'organisation du SMA en découlera directement, notamment pour la prise en compte des méthodes de communication. Cet état de fait nous permet d'inclure dans notre démarche les besoins d'une adaptation aux SE avant de finaliser nos agents, assurant ainsi un résultat plus adéquat.
IODA commence par établir les interactions nécessaires au système pour déduire les entités agent à créer. Une interaction est définie comme un bloc d'actions, impliquant un nombre fixe d'agents. Elle décrit quand et sous quelles conditions les agents vont interagir entre eux et avec leur environnement. Les attributs de ce bloc sont listés ci-dessous :
- **une condition de déclenchement** dont la description traduit implicitement un but ;
- **des pré-conditions** décrivant les dispositions logiques ou physiques nécessaires pour réaliser l'interaction ;
- **les actions** précisant les changements à appliquer sur l'environnement et les agents concernés ;
- **le nombre d'agents** impliqués. Ce nombre est représenté par une cardinalité (n,p) où n représente le nombre d'agents sources, et p correspond au nombre d'agents cibles ;
- **une liste de fonctions** exécutées par les différents agents impliqués dans l'interaction.

Cette méthode se focalise sur des cardinalités (1,1) ou (1,0). La première correspond à une interaction entre deux agents. La deuxième représente une interaction dite "dégénérée" réalisée par un agent source sur lui même. Dans ce dernier cas, la cible est représentée par un ensemble vide ;. A partir de la liste des activités de notre contexte, une matrice des interactions est construite pour obtenir une vue d'ensemble. Ensuite, le modèle IODA demande de spécifier les pré-requis de déclenchement. Pour ce faire, les auteurs

considèrent deux paramètres : la portée et la priorité. En effet, un agent source doit être à une distance suffisante pour initier l'interaction avec l'agent cible. De plus, si plusieurs interactions étaient simultanément éligibles, elles seraient alors sélectionnées selon leur priorité.

La Table 1 représente un exemple de matrice mis en avant dans les travaux de recherche liés au modèle IODA : le cas d'une fourmilière [19]. Nous y retrouvons les différentes familles d'agents impliquées par les interactions nécessaires à l'évolution de la fourmilière : une fourmi, la fourmilière, la source de nourriture, un morceau de nourriture et les phéromones des fourmis.

Nous pouvons constater que la famille d'agents la plus active est celle de la fourmi. Cette dernière initie à elle seule la majorité des interactions du système multi-agents. Vis-à-vis des agents de type "Source nourriture", elle peut initier trois interactions : les pister, ramasser un morceau de nourriture ou vérifier son chargement. La première interaction nécessite une distance minimale de 5 unités et possède une priorité de 6 (faible). Les deux autres interactions requièrent une distance de 0. Elles seront donc éligibles au même instant. La priorité permet d'établir une sélection : la fourmi vérifiera d'abord son chargement avant d'éventuellement prendre un morceau depuis la source de nourriture.

Nous retrouvons cette notion de priorité entre deux interactions éligibles chacune sur une cible différente. Par exemple, un morceau de nourriture initiera d'abord l'interaction "réveiller fourmi" avant d'envisager l'interaction dégénérescente "mourir".

Source \ Cible	∅	Fourmi	Fourmilière	Source nourriture	Morceau nourriture	Phéromone
Fourmi	+ (mourir, 8) + (vagabonder, 1) + (se-tourner, 0)		+ (suivre, 5, 6) + (stocker, 0, 5) + (se reposer, 0, 4) + (se décharger, 0, 3)	+ (pister, 5, 6) + (ramasser, 0, 5) + (déjàChargé, 0, 3)		+ (générer, 0, 7) + (suivre, 3, 2)
Fourmilière	+ (produireFourmi, 1) + (relacherEclaireur, 0)					
Source nourriture	+ (mourir, 0)					
Morceau nourriture	+ (mourir, 1)		+ (réveillerFourmi, 0, 0)			
Phéromone	+ (mourir, 1)					+ (disperser, 1, 0)

Table 1 – Matrice des interactions d'un SMA simulant le fonctionnement d'une fourmilière selon le modèle IODA [19]

Une fois les pré-requis établis, l'étape suivante consiste à lister les actions découlant de chaque interaction répertoriée, avant de les formaliser sous forme de fonctions à intégrer aux agents impliqués. Enfin, IODA formalise les architectures agents à partir de ces ensembles de fonctions en prenant en considération les points suivants :

- **une primitive d'initialisation** pour définir le démarrage de l'agent ;
- **un halo** déterminant comment l'agent perçoit les agents voisins ;
- **une matrice de mise à jour** pour décrire comment un agent évolue indépendamment des interactions

auxquelles il participe ;

- **les primitives des actions** correspondant à la matrice des interactions pour cet agent en tant que source. Ce modèle permet donc de rester indépendant du type d'agent et de son emplacement matériel lors de la spécification du SMA. C'est pourquoi nous avons développé nos architectures d'agents embarqués en nous inspirant de la matrice des interactions IODA.

2- Normalisation des échanges

Les systèmes multi-agents et les systèmes embarqués possèdent chacun leurs propres ontologies. L'ontologie représente une manière de décrire les informations d'un domaine. Au sein même des systèmes multi-agents, toutes les plates-formes ne possèdent pas la même ontologie. Or une représentation commune des données échangées est nécessaire pour établir une communication, a fortiori dans un contexte d'utilisation hautement hétérogène.

Pour définir une plateforme multi-agents embarquée, il est nécessaire d'établir une ontologie commune pour la mise en adéquation des messages des agents et des protocoles de communication embarqué. Ces derniers assurent la normalisation des échanges entre les processus et les différents SE. Il nous reste donc le choix d'une norme pour les communications multi-agents.

Deux propositions de standards existent :

- KQML (*Knowledge Query and Manipulation Language*) ;
- FIPA-ACL (*Foundation for Intelligent Physical Agents - Agent Communication Language*).

2.1 Norme KQML

KQML est un langage de haut niveau facilitant la communication entre agents logiciels. Développé dans les années 90, il fut récemment mis à jour pour répondre à des besoins d'abstraction de l'implémentation des systèmes multi-agents [20]. Orienté message, ce protocole emploie la théorie des actes de langage, c'est à dire l'association de chaque message à un moyen d'agir sur l'environnement de l'initiateur. Dans KQML, nous retrouvons cette notion à travers les performatives. KQML propose diverses performatives regroupées en 6 catégories. Nous présentons ces catégories dans la Table 2.

KQML n'engage pas l'utilisation d'un mécanisme de transport ni d'un langage déterminé et d'une ontologie spéci-fifiée pour le contenu d'un message. Il propose trois couches d'encapsulation pour un message. La première couche est celle du contenu, elle implique de préciser le langage du contenu ainsi que son ontologie. La couche suivante concerne le message dans sa globalité. Elle précise le type de message c'est à dire la performative, ainsi que les données du message. Enfin, la couche dite de communication détaille les informations concernant le transfert du message telles que le destinataire, l'expéditeur et l'adresse de retour en cas de réponse.

Compte tenu de nos besoins, l'application de ce standard nous pousserait à trouver une autre norme pour l'administration de notre plateforme. FIPA-ACL, pour sa part,est intégrée à une norme étendue à l'ensemble de la gestion d'un système multi-agents : FIPA.

Catégories de performatives	Description	Exemple
Les annonces (*Inform*)	Cette performative est utilisée pour agir sur la base de connaissance de l'interlocuteur en lui passant une information.	La performative "dire" (tell) permet à un agent initiateur d'indiquer quel service il possède à un agent destinataire.
Les requêtes (*Request*)	Lorsqu'un agent souhaite qu'un autre agent effectue un travail pour lui, il lui soumet une requête.	La performative "accomplir" (*achieve*) est utilisée pour demander à un autre agent d'effectuer une action donnée.
Les demandes (*Query*)	Semblables aux requêtes, les demandes impliquent un retour d'information de la part de l'agent destinataire.	La performative "demander tout" (*ask-all*) implique que l'agent destinataire renvoie toutes ses réponses à la question contenue dans le message.
Les réponses (*Reply*)	Après l'envoi d'une demande, une réponse est attendue et sera envoyée par le biais d'un message avec cette performative.	La performative "Envoyer tout" (*stream-all*) correspond à la performative de réponse à une demande de type "demander tout".
Les promesses (*Promise*)	A mi-chemin entre une information et une réponse, les promesses permettent d'indiquer à un instant donné la capacité d'un agent à effectuer une action.	La performative "avertir" (*advertise*) indique à un destinataire sa capacité à exécuter l'action du contenu.
Les flux (*Meta*)	Les flux sont des performatives d'indication et de mise à jour du SMA.	La performative "abonnement" (*subscribe*) indique à l'agent destinataire qu'il doit mettre à jour ses réponses à une performative donnée.

Table 2 – Catégories de performatives supportées par le protocole de communication inter-agents KQML[21]

2.2 Fondation FIPA et norme FIPA-ACL

La fondation FIPA propose l'établissement de standards dès 1996 pour favoriser l'utilisation d'agents logiciels dans des contextes industriels hétérogènes. Elle vise l'interopérabilité entre SMA développés par différentes compagnies et portés par différentes plateformes. En 2002, FIPA complète 25 spécifications [22]. Le standard propose tout d'abord la spécification d'une architecture abstraite ainsi que les relations entre les éléments composant cette architecture [spécification FIPA00001]. Elle regroupe également des guides d'instanciation d'agents basés sur l'architecture FIPA ainsi que des guides d'interopérabilité pour l'implémentation de FIPA. Cette spécification permet de décrire trois éléments en particulier :

- **l'administration** : une architecture de gestion du SMA dont la spécification reste abstraite pour permettre

une implémentation ouverte à tout langage de programmation ;

- **la communication** : la spécification des messages avec la norme FIPA-ACL et de nombreux protocoles d'interaction ;

- **le transport** : le routage des messages avec une couche d'abstraction appelée MTPS (*Message Transport Protocol Service*).

La norme FIPA propose une structure en plusieurs blocs. Nous présentons la hiérarchie de ces spécifications dans la Figure 3 [23]. Nous y retrouvons la notion d'une architecture générique avec des détails pou r l'administration de la plateforme, la communication entre les agents et le transport des messages. La couche d'applications fait référence aux agents logiciels développés par l'utilisateur de la plateforme multi-agents. Son positionnement en couche supérieure de la fifigure représente son appui sur l'architecture FIPA.

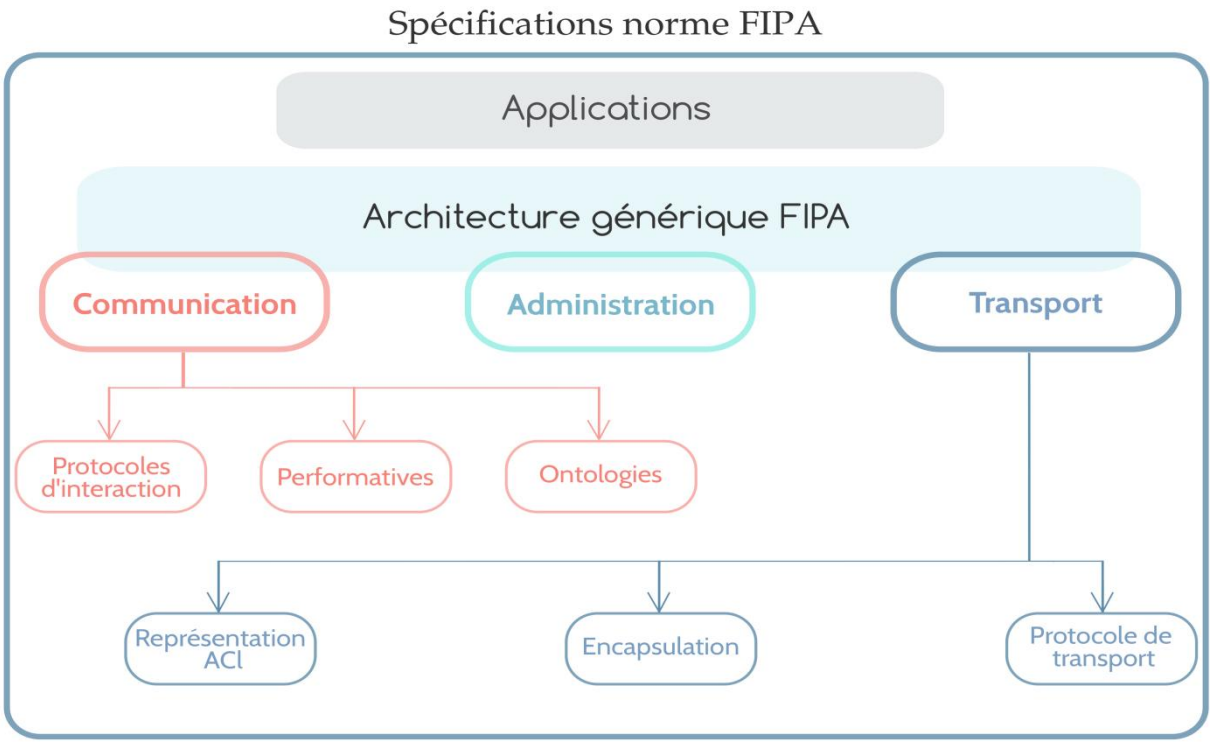


Figure 3 – Structure hiérarchique des spécifications FIPA.[23]

Nous présentons tout d'abord les spécifications liées à la communication avec le standard FIPA-ACL. Ensuite nous détaillons la partie administrative d'une plateforme multi-agents suivant la norme FIPA. Enfin nous décrivons les spécifications relatives au transport des messages.

FIPA-ACL est un standard de communication entre agents proposant une syntaxe similaire à KQML pour standardiser un ensemble de protocoles d'interaction [24]. Cette norme offre également une encapsulation des messages, bien que plus détaillée, et s'appuie sur le principe des performatives pour spécifier l'impact de ses messages.
A l'instar de KQML, FIPA-ACL est une norme indépendante du mécanisme de routage du message ainsi que du langage et de l'ontologie de son contenu.

Le standard regroupe une spécification de la structure des messages, un ensemble de performatives détaillant l'impact de chaque message et des protocoles d'interactions. Ces derniers précisent une suite d'échanges de performatives, formant ainsi une conversation complète, pour achever une forme spécifique d'interaction. FIPA-ACL compte actuellement 22 performatives de base qu'on peut regrouper en cinq catégories selon leur objectif global. Nous les détaillons en Table 4 [spécification FIPA00037].

Actuellement FIPA propose ainsi neuf protocoles d'interaction. Par exemple, le protocole d'interaction de requête (*FIPA Request Interaction Protocol*) est présenté en Figure 3.4 [spécification FIPA00026]. Nous y retrouvons un agent initiateur de l'interaction et un agent participant. Plusieurs agents peuvent être destinataires d'une interaction d'où l'utilisation du terme "participant". L'agent initiateur envoie sa demande par le biais d'un message possédant la performative *request*. Nous avons donc ici affaire à une demande d'action. L'agent participant va traiter la requête et prendre la décision de l'accepter ou de la refuser. Il notifie cette décision à l'agent initiateur par le biais des performatives *refuse* ou *agree*. Si la proposition est refusée, l'interaction se termine. Si l'agent participant a accepté la demande, il effectue la ou les actions liées à cette requête. En cas d'erreur, l'agent participant notifie l'agent initiateur du "non aboutissement" de sa demande par la performative de gestion d'erreur *failure*. Si tout s'est bien déroulé, l'agent participant emploie les performatives de passage d'information pour indiquer à l'agent initiateur la Complétion de la demande. Il utilisera *inform-done* si la demande ne nécessitait pas de retour d'information spécifique, *inform-result* sinon.

Enfin au niveau de la structure des messages, FIPA-ACL propose une encapsulation composée de 13 paramètres [spécification FIPA00061]. Parmi eux, 5 doivent être

obligatoirement complétés. Nous précisons leur fonction en Table3. Les autres paramètres représentent des champs de complétion optionnels et sont détaillés en Table4. La notion d'option, dans ce cadre, indique que le message doit tout de même comporter ce paramètre mais qu'il est possible de ne pas lui attribuer de valeur.

Catégorie	Description	Performatives
Passage d'information	Ces performatives sont utilisées pour transférer des connaissances parfois sous certaines conditions.	*inform, inform-if, inform-ref, inform-done, inform-result, confirm, disconfirm*
Demande d'information	Ces performatives sont utilisées pour faire des requêtes impliquant un transfert de certaines connaissances en réponse.	*query-if, query-ref, subscribe*
Négociation	Les performatives de négociation sont employées dans le cadre de protocoles d'interaction dédiés au principe de négociation.	*accept-proposal, cfp, propose, reject-proposal*
Demande d'action	Un agent peut demander à un autre agent d'effectuer une action spécifique parfois sous certaines conditions.	*request, request-when, request-whenever, agree, cancel, refuse*
Gestion d'erreurs	Deux performatives sont dédiées à la gestion des erreurs intervenant lors d'une interaction.	*failure, not-understood*

Table 3 – Performatives proposées par le standard de communication FIPA-ACL. [1]

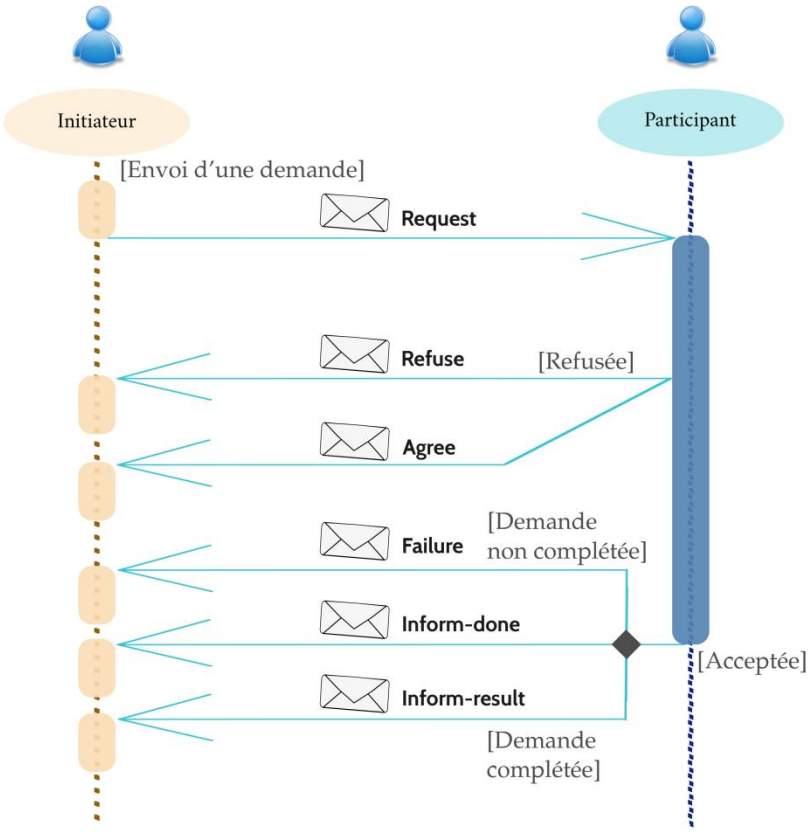


Figure 4 – Diagramme du protocole d'interaction FIPA dans le cadre d'une demande (request) [1]

Paramètre	Description
Performative	L'usage des performatives permet d'indiquer l'impact souhaité du message sur l'environnement de l'agent. Elle permet également un suivi lors de l'utilisation d'un protocole d'interaction défini.
Expéditeur (*sender*)	L'identité de l'expéditeur du message, c'est à dire son identifiant. Ce paramètre permet une réponse directe si besoin.
Destinataire (*receiver*)	L'identité du ou des destinataire(s) du message. Normalement ce paramètre est obligatoire, mais il est possible de le laisser vide lors de messages adressés à tous.
Répondre-à (*reply-to*)	Ce paramètre permet d'indiquer si une éventuelle réponse doit être renvoyée vers un autre agent. Si ce n'est pas le cas ce champ comporte la même information que le champ "expéditeur".
Contenu (*content*)	Ce paramètre correspond au message d'origine avant son encapsulation selon la structure FIPA-ACL. Les paramètres suivants intègrent des informations sur ce contenu.

Table 4 – Performatives obligatoires du standard de communication FIPA-ACL.[1]

2.3 Administration

La partie d'administration de la plateforme utilise les spécifications de FIPA-ACL. FIPA détaille un référencement des agents selon un identifiant AID (*Agent IDentififier*) permettant de distinguer chaque entité agent au sein du SMA [FIPA00023]. Cet identifiant se compose d'un nom et d'une "adresse", c'est à dire les informations nécessaires pour le contacter par message. Pour la gestion d'une plateforme multi-agents, la norme FIPA propose la mise en place de trois agents administratifs : AMS, DF et ACC [25].

• AMS

Le système de gestion d'agents AMS (*Agent Management System*) exerce le contrôle de supervision sur la plateforme. Un SMA ne doit pas comporter plus d'un agent AMS. Il tient une liste des agents présents et actifs du système multi-agents de manière semblable à la tenue d'un annuaire. Cette liste se complète au fur et à mesure grâce aux agents : lors de leur initialisation, ils informent AMS de leur disponibilité via leur nom et leur adresse. Lorsqu'un agent n'est plus actif il en informe également AMS et ce dernier le retire de sa liste. Ce service est assuré par deux fonctions d'AMS : l'enregistrement (*register*) et le dés enregistrement (*deregister*). Les agents peuvent également contacter AMS pour lui demander de mettre à jour les informations les concernant (statut ou adresse) par le biais de la fonction de modification (*modify*). Enfin, l'agent AMS met à disposition une fonction de recherche permettant d'obtenir l'adresse d'un agent donné (*search*).

Selon l'implémentation de la norme FIPA dans une plateforme multi-agents, AMS peut se voir octroyer des fonctions de contrôle supplémentaires. Par exemple, il est possible de lui permettre de demander à un agent de s'arrêter (*suspend agent*), voire de forcer son arrêt en cas de nécessité (*terminate agent*). Une autre option est de lui donner la possibilité d'ajouter des agents dans le système (*create agent*). Toutes ces possibilités doivent rester cohérentes avec la fonction principale de l'agent AMS : le contrôle du SMA.

Paramètre	Description
Langage (*language*)	Le langage est celui dans lequel le contenu du message a été exprimé. Si l'expéditeur est certain que le destinataire emploie le même langage que lui, ce paramètre n'a pas besoin d'être renseigné.
Encodage (*encoding*)	Si le contenu du message est encodé, ce paramètre sert à renseigner le destinataire sur l'encodage appliqué. Cette précision peut également être présente dans l'encapsulation de la couche de transport.
Ontologie (*ontology*)	La précision de l'ontologie du contenu est directement reliée au langage utilisé pour compléter sa compréhension. Si le langage n'a pas été spécifié ce paramètre n'est pas utilisé.
Protocole (*protocol*)	Si ce message s'inscrit dans une suite de messages déterminée par l'un des protocoles d'interaction proposé par FIPA-ACL, ce champ permet d'indiquer le protocole utilisé.
Identifiant de conversation (*conversation-id*)	Lors du suivi d'un protocole, tous les messages générés entre deux agents participants doivent posséder le même identifiant de conversation. Autrement, ce paramètre est optionnel.
Répondre-avec (*reply-with*)	Ce paramètre peut être employé lors d'échanges entre deux agents suivant plusieurs conversations en parallèle. Il permet de préciser le sujet de la conversation et fonctionne avec le paramètre suivant.
En-réponse-à (*in-reply-to*)	Ainsi dans le cadre de conversation parallèle, si un agent envoie un message avec une indication pour le champ "répondre-avec", l'autre répondra avec cette même indication pour le champ "en-réponse-à".
Répondre-avant (*reply-by*)	Ce paramètre indique un délai souhaité, en temps ou en date, à ne pas dépasser pour la réception d'une réponse de la part du destinataire.

Table 5– Performatives optionnelles du standard de communication FIPA-ACL[1]

• DF

Le facilitateur d'annuaire DF (*Directory Facilitator*) tient une liste des services actifs sur la plateforme et des agents auxquels ils sont reliés. Le maintien de cette liste permet de savoir à tout instant quelles capacités sont à disposition sur la plateforme. Il est possible d'avoir plusieurs agents DF au sein d'un SMA. La répartition des capacités sur les différentes listes peut alors se faire selon leur type, ou bien selon leur localisation si le SMA s'étend à plusieurs machines. D'autres critères de répartition peuvent être choisis.

A l'instar d'AMS, DF propose des fonctions d'enregistrement, de dés enregistrement et de modification des informations sauvegardées. Les agents l'informent de leurs capacités lors de leur activation et se dés enregistrent à leur terminaison. L'agent DF apporte également une fonction de recherche pour obtenir l'identité des agents proposant un service donné. Lorsque plusieurs agents DF sont présents sur un même SMA

et que la fonction recherche de l'agent DF contacté ne donne pas de résultat, il va étendre cette recherche aux autres agents DF du système.

Il est important de noter que l'agent DF ne fait qu'indiquer quel agent fournit quel service. Il ne peut garantir que l'agent ensuite contacté sera en état de réaliser ce service à un instant donné. Nous résumons ce point par le fait que l'agent DF offre une liste des capacités existantes, mais pas forcément une liste des capacités disponibles. Une option de délai de validité d'une capacité peut être mise en place pour s'approcher d'une liste à jour. Cette option implique cependant que les agents reviennent régulièrement informer DF des capacités qu'ils proposent, sans pour autant assurer qu'ils seront en état de la réaliser lorsqu'ils seront contactés. Elle reste pertinente dans le cadre d'une longue période d'exécution.

• **ACC**

Enfin, le canal de communication entre agents ACC (*Agent Communication Channel*) fournit un service fiable et précis pour le routage des messages à destination d'autres agents. Il peut y avoir plusieurs agents ACC au sein d'un même système multi-agents. La capacité de l'agent ACC permet aux agents logiciels composant la plateforme de ne pas devoir enregistrer les informations des uns ou des autres. Lorsqu'un agent veut en contacter un autre, il envoie son message à ACC avec une "enveloppe" contenant les informations nécessaires à l'identification du destinataire. Le message à transférer contient au minimum les champs de performative, d'expéditeur, d'agent à contacter en retour et du contenu avec les diverses options y étant rattachées. L'agent ACC utilise ensuite les fonctions de recherche des agents DF et AMS pour compléter les paramètres du message à transférer à partir des informations confiées dans l'enveloppe. Enfin il envoie le message complété au destinataire d'origine. Cette gestion relève de l'encapsulation, la diffusion de l'ensemble des messages du système à travers la couche de transport est gérée par la couche MTPS.

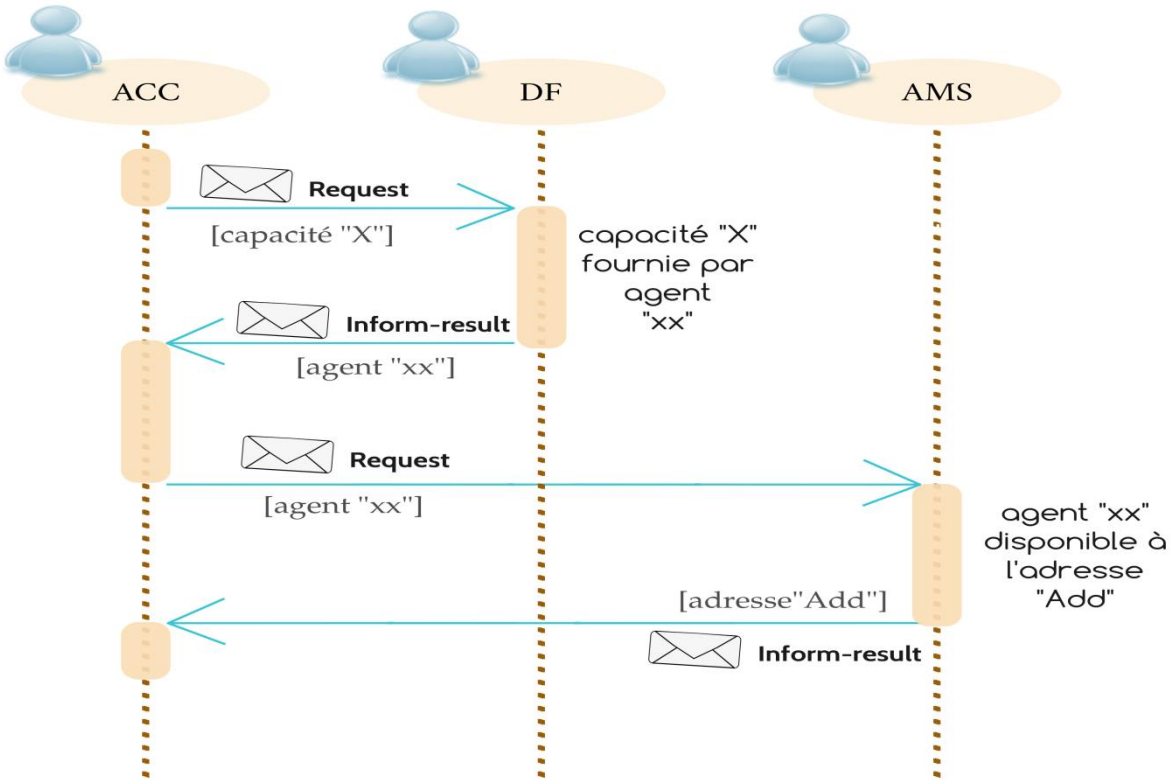


Figure 5 – Diagramme d'échanges entre les agents ACC, DF et AMS pour compléter les paramètres d'un message confifié à ACC pour être transféré.[25]

La récupération des informations auprès des agents AMS et DF implique des requêtes. Il est préconisé de suivre le protocole d'interaction de demande, présenté en section 2.2, mais ce n'est pas une obligation. En Figure 5 nous présentons un exemple d'interaction entre les agents ACC, DF et AMS visant à compléter les paramètres d'un message. Dans le cadre de cet exemple, l'enveloppe envoyée à ACC contenait l'indication de la capacité "X" du destinataire. L'agent ACC utilise donc d'abord la fonction de recherche de l'agent DF pour obtenir l'identification de l'agent proposant cette capacité. L'agent DF traite la demande en parcourant sa liste, il renvoie ensuite l'identifiant de l'agent fournissant ce service : "x". Avec cette information, l'agent ACC peut ensuite contacter l'agent AMS afin d'obtenir l'adresse "Ad" qu'il pourra indiquer en paramètre "destinataire" du message à transférer. Dans cet exemple, la complétion des paramètres implique la génération de quatre messages. Si on choisit de suivre le protocole d'interaction d'une demande, le nombre de messages passe alors (dans le cas d'une demande complétée) à un total de six. L'envoi concret du message est ensuite passé à la couche de transport de la plateforme.

2.4 Transport des messages

FIPA est un standard indépendant du protocole de transport utilisé par la plateforme multi-agents l'appliquant. Elle propose tout de même un guide d'architecture pour favoriser l'interopérabilité à travers la spécification du protocole de transmission des messages MTPS [FIPA00067]. Ce protocole spécifie la nécessité d'appliquer une abstraction entre la notion de transport et le protocole effectuant ce transport. Au niveau de la gestion de l'administration, c'est l'agent ACC qui se charge de la gestion de la notion de transport en préparant l'encapsulation complète d'un message pour son transfert. L'application concrète du transport est gérée par la couche MTPS. Cette couche analyse les paramètres d'encapsulation et utilise les protocoles de transport à sa disposition pour effectuer le transfert concret. Par exemple, la couche MTPS peut considérer les champs d'expéditeur et de destinataire pour vérifier les adresses indiquées par ces champs. Si les adresses renvoient à la même machine, le protocole de transport utilisé pourra être différent d'un cas d'envoi entre deux machines différentes. Ce choix reste opaque pour les agents logiciels et administratifs. Par ailleurs, si l'encapsulation du message nécessite de nouveaux paramètres en fonction du protocole de transfert utilisé, c'est également le rôle de la couche MTPS d'assurer la correspondance de la structure du message.

Le standard FIPA nous permet de normaliser la communication de nos agents. Les algorithmes d'interactions employés par ces agents relèvent de leurs comportements sociaux.

3 - Algorithmes de comportements sociaux multi-agents

Les systèmes multi-agents ont de fortes capacités d'adaptation et d'auto-organisation leur permettant de former des systèmes hétérogènes évolutifs. Régulièrement, du fait de l'exécution simultanée des actions des différents agents, des relations synergiques ou conflictuelles peuvent émerger. La nature de ces relations dépend de la compatibilité des buts des différents agents, de leur capacité à accomplir ces buts et du nombre de ressources disponibles pour cette réalisation [2].

Nous pouvons ainsi déterminer le type de comportements sociaux qu'adopteront les agents en suivant l'arbre de décision présenté en Figure 6.

La compatibilité des **buts** est éprouvée par les actions nécessaires pour achever ces derniers. Si les actions sont complémentaires, les buts seront déterminés comme compatibles. Si, à l'inverse, les actions entrent en conflit, ils sont déterminés comme

incompatibles. Par exemple, si le but d'un agent est de sortir d'une pièce fermée, il devra en ouvrir la porte. Si le but d'un autre agent est d'empêcher quelqu'un d'entrer, il devra s'assurer que la porte reste fermée. Les buts de ces deux agents sont incompatibles car ils créent un conflit sur les actions relatives à la porte.

Nous évaluons ensuite les **ressources** considérées. Sont-elles suffisantes pour l'ensemble des agents ou impliquent elles une gestion supplémentaire ? Par exemple, pour nos deux agents ayant des buts incompatibles, les ressources liées à leurs buts sont les portes. Si il n'y a qu'une seule porte les agents entreront obligatoirement en conflit, la ressource est alors "Insuffisante".

Enfin, nous estimons les **compétences** des agents indépendamment de l'état de compatibilité des buts et des ressources. Nous devons pour ce faire vérifier si les agents ont ou non les capacités d'effectuer toutes les actions nécessaires à l'accomplissement de leurs buts. Par exemple, si nous reprenons l'exemple lié aux buts, nous vérifierons que le premier agent a la capacité d'ouvrir la porte pour sortir. Nous évaluerons également la capacité du deuxième
agent à pouvoir détecter l'ouverture de la porte et savoir la refermer si besoin.
Suivant l'établissement de ces trois facteurs, nous pouvons déterminer la nature des relations sociales qui auront cours au sein du SMA. Si les buts sont compatibles, les ressources et les compétences suffisantes, alors le SMA sera composé d'agents totalement indépendants n'ayant pas besoin d'établir de relations sociales particulières.

Dans le cadre de buts compatibles mais de ressources insuffisantes ou de compétences insuffisantes, les agents seront dans une optique de coopération pour atteindre leurs objectifs. Les interactions peuvent alors établir cette coopération et sont qualifiées d'interactions de coordination [26] . Pour des buts incompatibles mais des ressources suffisantes, des relations de compétition émergeront, sensiblement différentes selon les compétences des agents. Si les ressources sont également insuffisantes les relations seront conflictuelles. Les agents doivent alors procéder à des interactions pour résoudre le problème posé tout en satisfaisant l'intérêt respectif de chaque agent, c'est à dire qu'un accord mutuellement acceptable est recherché. Ces interactions sont qualifiées de négociation .[27]

3.1 Coordination et négociation

La coordination des agents peut être considérée comme un ensemble d'actions supplémentaires à accomplir en coopération avec les autres agents du SMA. Ces actions correspondent à une programmation, d'une part de l'utilisation des ressources communes, d'autre part d'un enchaînement de compétences pour permettre l'accomplissement des buts compatibles.

Pour établir cette programmation, il existe quatre types de coordination : par synchronisation, par planification, par réglementation et par réaction. La synchronisation correspond à des protocoles généralement implémentés à l'avance. Les agents échangent de nombreuses informations de manière à détecter les conditions d'application du protocole de synchronisation. La planification peut être la charge d'un seul agent qualifié de superviseur. Autrement, elle est distribuée entre les différents agents, chacun proposant une planification, avec une mise en commun permettant de faire émerger une entente globale. Cette dernière se base sur des "promesses d'action" de la part des agents. Le principe de réglementation correspond à un ensemble de règles connues par tous les agents leur permettant de savoir quel comportement adopter vis-à-vis des autres à l'instar d'un code de la route par exemple. Enfin, une coordination réactive permet de s'adapter à des agents eux-même réactifs avec un principe de stimuli/réponse.

En termes de temps de réponse, seule la coordination par planification peut être qualifiée de "lente". Elle est cependant la meilleure solution en termes de prédictibilité, c'est à dire d'organisation à long terme. La synchronisation ne représente un avantage que pour sa rapidité à obtenir une solution [28]. Cette dernière reste efficace sur le court terme et laisse la place à peu d'adaptabilité. Quant à la coordination par règlement, il s'agit d'une méthode relativement efficace offrant un bon compromis.

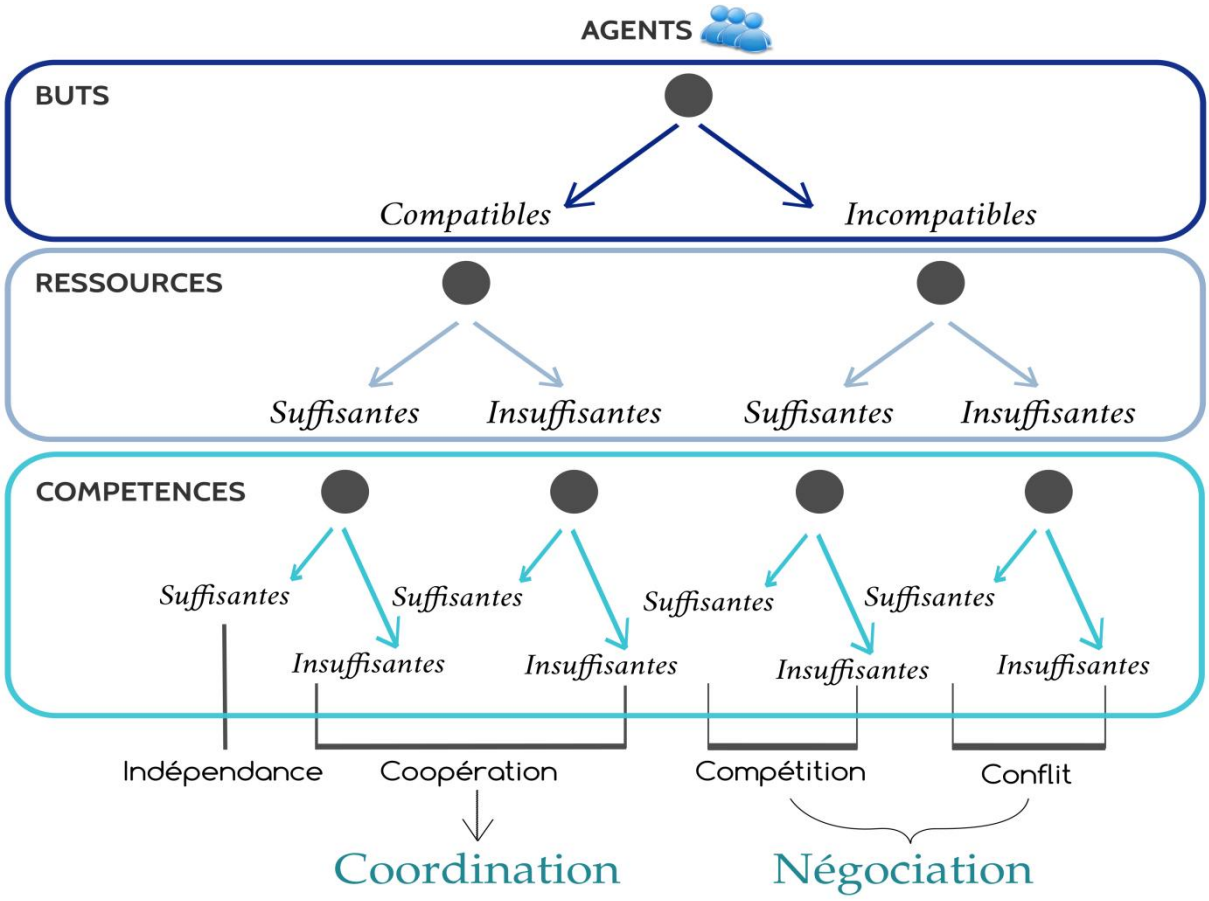


Figure 6 – Diagramme de détermination des comportements sociaux en fonction des buts, des ressources et des compétences des agents concernés.[1]

Les protocoles de négociation et de coordination des systèmes multi-agents permettent de faire émerger une solution optimale en peu de temps, dans des environnements dynamiques

Conclusion :

Dans ce chapitre nous avons présentées une vision globale des systèmes multi agents et des agents. Ces systèmes c'est un réseau d'agent qui interagissent, communiquent et coopèrent entre eux pour accomplir un objectif bien précise ; et nous avons décrivons tous les éléments constituants un SMA on part de l'agent à l'organisation en passant par l'environnement et finalement, les interactions et communication entre agents.

Chapitre 02:

Négociation multi-agents

Introduction :

Dans le cas des SMA, où les agents sont autonomes, la négociation est la meilleure solution pour résoudre les conflits entre eux.

Alors la négociation est un mécanisme inspiré du modèle humains pour résolution des conflits ; ainsi qu'elle permet à tous les agents d'atteindre leurs buts et arriver à un accord commun pour les deux parties négociateurs. Dans ce chapitre on va détailler qu'est-ce qu'une négociation, ses composants de base ainsi que les modèles et les formes de négociation automatique ; dans le deuxième partie de ce chapitre on va présenter la négociation multi issue.

1. La négociation :

La négociation est toujours un sujet très intéressant dans les SMA. Selon le point de vue de MH.VERRON dans [29], il y a négociation lorsqu'il y a une discussion, des propositions entre les protagonistes et lorsque l'accord final satisfait au mieux tous les participants. VERRON a donné aussi une autre définition dans [29] : « Une négociation met en jeu des ressources, qui seront rassemblées afin d'être négociées dans un contrat et un ensemble de personnes qui participent à cette négociation. Il y a toujours un ou plusieurs manageurs (vendeur ou autre) et un ou plusieurs contractants (acheteurs ou autre) ».

Autre définition ; une négociation est un octuple (G, R, P, L, S, G, H et Ct) où :

- **G** est l'ensemble des individus impliqué dans la négociation.

- **R** est l'ensemble des ressources disponibles.

- **P** est un ensemble de relation de préférence local de chaque agent.

- **L** est l'ensemble des actes de langage autorisé pour la négociation.

- **S** est un ensemble de règles de séquencement.

- **G** est un graphe dans les nœuds sont des règles de séquencement.

- **H** est l'histoire de la négociation.

- **Ct** est un contrat qui est l'objet de la négociation. [30]

Pour assurer une négociation réussite, La modélisation des processus doit prendre en compte plusieurs Aspect. Ces aspects doivent être clairement spécifiés :

Le langage de négociation : comme on a vu précédemment que la communication est le support de l'interaction. les agents ne peuvent pas négocier sans échange des messages.

Autre que ça, le langage de communication doit inclure l'expression primitive de communication. Par exemple ; en utilisant le langage de communication, un agent doit être capable d'exprimer son avis concernant une proposition (acceptation ou refuse) comme il doit être capable de

proposer ou de refuser la participation à une négociation [31].

- **Le protocole de négociation :** le protocole de négociation précise les règles de rencontre entre les agents négociateurs. Ces protocoles spécifie zpour chaque moment du processus de négociation un ensemble valide de réponses ou d'interactions. Par exemple, dans Dès réception d'une demande de participation à la négociation, l'agent a droit de participer ou de refuser

la participation. C'est inacceptable et non valide d'envoyer un message d'échec à l'initiateur à ce niveau. En plus, les protocoles déterminent la terminaison de négociation avec réussie ou terminaison échouée.

- **L'objet de négociation :** il faut spécifier clairement l'objet de négociation avant le début de processus. L'agent sera une entité perdue et sans objectif.Un agent vendeur qui minimise le prix d'un produit « X » pour arriver à un accord avec un agent qui cherche un produit « Y » et qui n'est pas intéressé par le produit « X » est un agent irrationnel. On note que l'objet de la

négociation peut être spécifié par un seul attribut (par exemple, le prix) ou plusieurs attributs (comme le type de produit, la couleur, la qualité et le prix) [31].

- **Le processus de décision :** Bien sûr, les décisions d'agent durant le processus de négociation ne sera pas une décision aléatoire. Les agents doivent suivre une stratégie spécifique au cours de ce processus appelée **stratégie de négociation.** Cette stratégie permet aux agents de prenez les bonnes décisions à tout moment. Par exemple, un acheteur qui négocie pour acheter un produit avec le moindre prix

possible, quand est-il décide que le prix proposé par le vendeur à un instant donné est le prix minimum ? Pour prendre ces décisions, l'agent doit être capable de raisonner sur le raisonnement des autres agents. Il doit être aussi capable d'identifier les stratégies suivies par les autres agents. Un joueur d'échec ne raisonne pas seulement sur ses mouvements possibles mais aussi sur les mouvements possibles de son adversaire [31].

- **La cardinalité des participants de la négociation :** négociation un-à-un,un-à plusieurs ou bien plusieurs-à-plusieurs. [32]

2. Les formes de négociation :

Il y a plusieurs techniques de négociation on peut les traiter dans le domaine des SMA, dans cette partie on va citer quelque formes de négociation :

2.1.Les systèmes de vote :

Le système de vote est utilisé pour choisir un suppléant parmi les différents électeurs. Le plus simple consiste à choisir entre des alternatives et des alternatives Statu quo. Il s'agit de proposer des alternatives et de recueillir les votes et les bulletins de vote contre ce choix. Les systèmes plus complexes impliquent de nombreux plus de deux alternatives, alors les électeurs doivent choisir l'alternative qu'ils choisissent comme ça. Le terme choix social est utilisé pour indiquer un choix qui remplit une condition Le plus peuplé.Les systèmes de vote sont des systèmes de négociation à un seul tour de parole (les participants envoient leur liste de préférences et l'initiateur réalise la procédure choisie) qui permettent de choisir parmi toutes les alternatives possibles. Il faut donc que chacun évalue l'ensemble des alternatives et les classe par ordre de préférence avant que le vote n'ait lieu. Cela est alors très coûteux en temps et en espace. « *GeNCA : un modèle général négociation de contrats entre agents* ». [33]

2.2.Les enchères :

La vente aux enchères est un mécanisme de négociation où les agents rivaliser avec plusieurs autres agents pour acheter ou vendre des biens. Cette concurrence permettra de déterminer objectivement le prix d'un objet. Ce mécanisme est bien connu depuis l'Antiquité, de plus il est utilisé dans différent site de vente en ligne.

La philosophie générale d'enchère est la proposition des prix par des participants. Ensuite, l'initiateur va décider la meilleure proposition. Bien entendu, le nombre de tours (c'est-à-

dire le nombre de fois que l'agent peut proposer un prix), la méthode d'annonce des propositions (publique ou privée) et la méthode de choix de vainqueur sont les caractéristiques qui font la différence entre les protocoles d'enchère [31]. Cependant, il y a des choses en commun dans la plupart des protocoles, par exemple s'il existe un prix de réserve, ce prix est le plus donnée par l'initiateur pour un objet. Par conséquent, les participants n'ont pas le droit d'annoncer en dessous du prix de réserve. Si tous les participant fournir le prix au prix de réserve le protocole va terminer par un échec.Dans le domaine des SMA, il existe quatre catégories principales d'enchères :

Les enchères ascendantes (Anglaise) :

Ces enchères appellent ascendante car la stratégie de ce protocole est d'augmenter les prix avec le temps. Elles sont aussi dites enchères ouvertes, orales ou anglaise. Ce protocole s'applique aux tours successifs, ou le prix est successivement augmenter jusqu'il reste l'agent vainqueur, celui qui propose le plus grand offre et gagner l'article.

Autrement dit, il y a une discussion entre le vendeur et les acheteurs, ce qui est bien à notre sens de négociation. Cette négociation peut se formaliser comme suit : le vendeur commence l'enchère par l'annonce de prix de réservation, ensuite les autres agents(les acheteurs) vont annoncer publiquement un prix plus grand que le prix proposé dernièrement, les agents continuer la proposition des prix jusqu'à l'obtention d'un prix gagnant. Un prix est gagnant si aucun agent ne peut augmenter par rapport àce dernier. Le protocole est terminé, l'acheteur doit acquit l'objet, il ne peut plus le rendre .

Les enchères descendantes (Hollandaises) :

Ces enchères dites descendants car le prix va diminuer dans chaque tour. Elles sont utilisées notamment aux **Pays-Bas** pour la vente des fleurs, c'est pourquoi les économistes les appellent aussi enchères hollandaises .

Ces enchères est considérer aussi comme une négociation car elles se déroulent sur plusieurs tours. Le protocole se définit comme suit : le vendeur présente leur article avec un prix élevé à un ensemble d'acheteurs ; si l'acheteur accepte la proposition il aremporé le contrat. Si tous les acheteurs refusent, le vendeur va poser leur article avec un prix inférieur. Le processus se termine quand l'acheteur accepte la proposition, ou lorsque le prix de réserve est atteint et que personne n'accepte la proposition.

- **Les offres scellées au premier meilleurs prix :**

Ces enchères se déroulent sur un seul tour où les acheteurs proposé un prix une seule fois « prix unique » sans connaitre les propositions des autres agents, c'est pour cela elles dites offres scellées. Le vainqueur ici celui qui propose le plus élevé prix et le paie.

Comme les acheteurs ne peuvent plus faire une deuxième tour, ces enchères ne sont pas des négociations au sens strict. Alors le protocole est très simple : le vendeur propose un article à un ensemble des acheteurs. Ceux-ci peuvent soit accepter la proposition où il propose un prix, soit refuser la proposition. Le gagnant de l'enchère celui qui propose le plus élevé prix (supérieur au prix réservé). L'une des différences avec les précédentes enchères est qu'il n y a pas de contre proposition possible .

- **Les offres scellées au second meilleur prix (vickrey) :**

Ce protocole est similaire au protocole de premier meilleur prix, mis à part que l'agent gagnant paie le deuxième prix proposée. Ici aussi, il n'y a pas une seconde proposition possible implique y a pas une négociation proprement dites. Le protocole est identique à celui des offres scellées au meilleur prix, la seule différence étant le prix à payer par le gagnant du contrat [34].

2.3.La négociation à base d'argumentation :

La négociation à base d'argumentation est utilisée chez les agents logique squi possèdent une base de connaissance avec des prédicats et des règles d'inférences [35].l'argumentation est donc conçu pour changer les croyances des autres agents afin qu'il adopte le même point de vue, les même croyances etintensions que l'agent argumentant.

Au temps de négociation les agents peuvent utilisés plusieurs types d'arguments ; ainsi que chaque argument à leur propre précondition pour l'utilisé.

L'agent doit choisir le bon argument et l'utiliser lorsqu'il est satisfait, c'est pour cela l'agent a besoin d'une stratégie pour effectuer leur choix. A partir de qui ce précède on trouve les types d'arguments suivants :

Appels à une promesse passée :

Le négociateur A rappelle à B une promesse passée concernant l'objet de négociation. Autrement dit l'agent B a promis dans une négociation antérieure à l'agent A d'offrir ou effectuer un objet de négociation.

- Préconditions : l'agent a doit vérifier si une promesse concernant un objet de négociation a été reçue dans le passé à l'occasion d'une négociation conclue avec succès.

Promesse d'une récompense future :

Le négociateur A promet de faire l'objet de négociation pour un autre agent B à un moment futur.

- Préconditions : l'agent A doit trouver un désir de l'agent B pour un moment futur, si possible un désir qui peut être satisfait par une action (service) que A peut effectuer mais que B ne peut pas effectuer.

Appels au propre intérêt :

L'agent A croit que la conclusion d'un accord sur l'objet de négociation est dans l'intérêt de B et essaye de convaincre B de cela.

- Préconditions : l'agent A doit trouver (ou déduire) un des désirs de B qui sera satisfait si l'agent B a l'objet de négociation ou A doit trouver un autre objet de négociation qui a été offert au par avant sur le marché et démontrer que l'objet de négociation proposé est plus intéressant que

Appel à une pratique fréquente :

L'agent A croit que B a refusé la proposition parce que B croit que la proposition contredit un de ses buts. Dans ce cas, l'agent A donné à B l'exemple d'une pratique fréquente qui démontre que l'acceptation de la proposition ne contredit pas le but de B.

- Précondition : l'agent A doit trouver un autre agent ayant le même but que B, qui a été déjà d'accord avec une proposition semblable et qui a vu que l'accord n'a pas nui à ses buts.

Menace :

Le négociateur menace de refuser à faire/offrir quelque chose à B ou il menace de faire quelque chose qui contredit les désirs de B.

- Préconditions : l'agent A doit trouver un des désirs de B directement satisfiable par un objet de négociation que l'agent A peut offrir ou A doit trouver une action qui est contradictoire avec ce qu'il croit être un des désirs de B [35].

3. Les protocoles de négociation :

3.1.Définition

Un protocole de négociation n'est pas loin d'un protocole d'interaction,seulement que la négociation nécessite un conflit entre les agents. Alors nous adoptons la définition suivante : « Chaque forme de négociation possède son propre protocole, qui définit le déroulement du processus de négociation, c'est-à-dire les actes de langage utilisés et leur séquencement » 31]

3.2.Classification des protocoles de négociation :

Dans cette section on va citer quelques propositions de classifications des protocoles de négociation :

3.2.1. Les protocoles compétitifs vs coopératifs :

Selon P. Maes, il identifie deux classes de protocoles.

La première classe regroupe les protocoles distributifs qui se basent sur le principe du partage des gains entre les participants. Chacun des participants cherche à maximiser sa part de profit, la réussite de l'acheteur signifie la défaite du vendeur,et vice versa.

Dans la théorie de jeu, on les appelle des jeux à somme nulle. P. Maes les considère comme des protocoles compétitifs, car c'est l'esprit qui domine dans ce genre de protocoles. Les protocoles de vente aux enchères sont considérés comme des protocoles compétitifs.

La deuxième classe regroupe les protocoles intégratifs, dont l'objectif de ses participants est d'essayer d'élargir l'espace d'entente, et de chercher de nouvelles solutions qui maximisent les profits mutuels des différents participants. Dans la théorie de jeu, on les

appelle des jeux à somme non nulle, ils sont considérés comme des protocoles coopératifs en intégrant plusieurs paramètres de négociation. [33]

3.2.1. Les protocoles unidirectionnels vs bidirectionnels :

Dans les protocoles unidirectionnels, nous sommes limités à accepter ou rejeter la proposition. Cette catégorie comprend les protocoles de ventes aux enchères et Contract Net.

D'autre coté les protocoles bidirectionnel (son scénario sont plus complexes) est basé sur l'échange des messages (offre et contre-offre). Ce type de protocoles fournit plus négociation flexible on modifiant d'un ou plusieurs attributs peut être négocié via une contre-proposition ou bien nous changeons la structure de la proposition. Par exemple nous ajoutons de nouveaux attributs négociables.

Il y a d'autre classification pour les protocoles de négociation, sont les suivants :

Wurman et al, proposent une classification basée sur les critères suivants :

Simple ou double : un protocole est dit simple si le type des enchérisseurs est soit des agents acheteurs ou soit des agents vendeurs. Dans la double enchère on trouve des agents acheteurs et des agents vendeurs qui participent à la même enchère.

Privé ou public : dans le premier cas, les propositions soumises par les participants ne sont connues qu'après la clôture de l'enchère. Dans le deuxième cas, les propositions sont publiques, connues par tous les participants.

Conditionné ou inconditionné : Un protocole est conditionné si les propositions sont régies par des règles, par exemple, le prix proposé doit être ascendant ou des cendant.

London Classification est un modèle de classification des systèmes de négociation,défini à l'issue du workshop *Négotiations in Electronic Markets*. Il traite l'aspect participant, l'aspect produit, l'aspect décisionnel et l'aspect processus de négociation. Ce dernier aspect est défini par un ensemble d'attributs, parmi les quels :

Le type d'agent initiateur de la négociation qui peut être l'agent vendeur ou l'agent acheteur.

Le nombre de phase : un protocole est dit à simple phase si les règles de négociation ne changent pas tout au long du processus de négociation, dans le cas contraire,nous le considère comme un protocole multi-phases

L’ordonnancement dans la négociation multi-attributs peut être une seule étape sitous les attributs sont négociés en même temps ou multi étapes si on négocie chaque attribut à part.

La publication des offres ou des propositions, si la proposition est entendue par tous les participants le protocole est dit public (ou open cry pour les protocoles de venteaux enchères).Dans le cas contraire on l’appelle protocole privé ou sealed bid.

Le prix que paie le gagnant dans l’enchère dépend des règles du protocole utilisé :dans les protocoles discriminatifs le gagnant paie le prix de la proposition qu’il asoumis (par exemple l’enchère anglaise). Dans le protocole non discriminatifs le gagnant paie le prix plus bas que celui qu’il avait proposé. Par exemple, dans le protocole Vickrey le gagnant paye le prix de la deuxième proposition.

Le désistement d’un participant sur un engagement qu’il avait pris au cours d’une négociation peut être permis ou non par le protocole de négociation.

Sandholm aussi identifie cinq types de protocoles de négociation : les systèmes de vote, les enchères, marchandage, systèmes contractuels et la formation de coalition.

3.2. Quelques protocoles de négociation :

Il existe plusieurs protocoles de négociation, on va parler sur les deux protocoles les plus populaires :

Le protocole Contract Net :

Le Protocole *Contract Net* proposé par Smith en 1980 est un mécanisme de négociation entre deux types d'agents : contractant et gestionnaire [34]. Le Contract Netpermet après quelque discussion avec un groupe d’agents, le gestionnaire conserve le service d’un agent qui s’appelle un contractant pour effectuer une tache (contact). Le protocole est qualifié « sélection mutuelle » car pour « signer un contact », l’agents électionné doit s’engager pour l’exécution de la tâche et le gestionnaire ne sélectionne que l'agent Fournit les propositions les plus favorables. La version originale du protocole comporte trois étapes principales :

1. L'appel d'offre

2. La soumission de proposition

3. L'attribution de contrat.

La section suivante présente l'algorithme de protocole Contract Net détaillée en septétapes ; regroupe les traitements du gestionnaire et des contractants, ainsi que les échanges entre eux.

Pour appliquer ce protocole, le concepteur doit détailler le contenu des messages échangés, le temps d'expiration ainsi que les fonctions « évalue-annonce » et «évalues ou mission » [34].

Etant donné une tache, un gestionnaire, un groupe de (n-1) soumissionnaire :

1- le gestionnaire envoie un message de type « annonce- tache » à un groupe d'agents (ou fait un 'broadcast').

2- Chaque agent évalue la tache annoncée à l'aide d'une fonction locale 'évalue annonce'.

3- L'évaluation précédente permet à certains agents de soumettre une proposition à l'aide d'une 'soumission-tache' au gestionnaire.

4- Si une proposition est jugée satisfaisante (l'aide de la fonction 'évalue soumission'), alors le gestionnaire envoie un message de type 'acceptation' à celui dont la proposition est retenue. Il envoie également un message de type 'refus' aux autres agents dont les propositions n'ont pas été retenues.

5- Le gestionnaire peut mettre fin à la période d'acceptation de proposition si le temps d'expiration est dépassé.

6- C'est aucune proposition n'a été retenue, alors le gestionnaire fais parvenir à tous les agents non retenus un message de type 'refus' pour indiquer le rejet de chacune des propositions.

7- Il peut alors se retirer de la négociation, retenir la proposition la plus acceptable redémarrer un nouvel appel d'offre (nouveau 'annonce- tache') ou prolonger le temps d'expiration de la période d'acceptation de proposition.

L'agent ayant obtenu le contrat, remet un rapport d'exécution lorsque la tâche est complétée 34].

Le protocole Kasbah : a été développé au MIT Media Lab par Pattie Maes en 1996.C'est un système où les utilisateurs créent des agents pour négocier la vente et l'achat de biens pour leur compte sur Internet. Ces biens sont classifiés, reprenant ainsi l'idée des petites annonces classées par type, voici un exemple de négociation avec le protocole

Kasbah [31]. Lors de la création d'un agent pour acheter ou vendre,l'utilisateur spécifie le type d'attribut Négocier, la date à laquelle il veut trader, le prix attendu, le plus Petit (ou grand) prix acceptable et stratégie de négociation choisis parmi 3 propositions Correspond aux fonctions linéaires, quadratiques et

exponentielles pour le calcul Les prix changent au fil du temps. L'utilisateur spécifie également si l'agent doit demander Consentement avant de conclure un accord et si vous souhaitez être averti par e-mail lorsqu'un accord est conclu. Une fois un accord conclu, des transactions physiques peuvent avoir lieu et doivent être fabriqué par des agents humains.

4. Évaluation des protocoles de négociation :

On doit choisir le meilleur protocole pour chaque système ou domaine d'application ; pour assurer le choix d'un bon protocole il faut être capable d'évaluer les protocoles proposés, cette dernière effectuée seulement si on suit les critères d'évaluation suivants :

Rationalité individuelle : on dit qu'un protocole est rationnel pour un agent en temps qu'il participe à la négociation ; il garantir un gain précise. Ou bien dit,l'agent ne perd rien en participant à la négociation [30]. Un agent apparemment cherchera à travers la participation à la négociation à accroitre leur fonction d'utilité.

Donc si la participation aux négociations n'augmentera pas les revenues d'agent, il n'y a pas un besoin pour participer.

Le bien-être social : Parfois, nous pouvons choisir de négocier un protocole Non pas parce qu'il est rationnel envers l'agent, mais parce qu'il est le meilleur par rapport à la société. Les fonctions de choix social agrègent les utilités des individus.

Alors que maximiser le bien-être utilitaire (la somme des utilités) vise à satisfaire la société dans son ensemble, maximiser le bien- être égalitaire (l'utilité minimale)consiste à réduire les inégalités dans la population, i.e. améliorer la satisfaction de l'agent le moins bien doté. Maximiser le bien-être de Nash (le produit des utilités)est un compromis entre ces deux règles de décision collective. Une alternative qui maximise le bien-être utilitaire ou le bien-être de Nash est optimale au sens de Pareto [35]. Par conséquence, nous comparerons les protocoles par rapport aux critères d'efficacité globale.

L'efficacité Pareto : Une solution (résultat) x à un problème de négociation est efficace Pareto (Pareto optimale), s'il n'y a pas une autre solution x' telle qu'au moin sun agent ait une meilleure situation avec x' qu'avec x et aucun agent n'ait une plus mauvaise situation avec x' qu'avec x [36]. En d'autres termes, s'il n'y a pas d'autre résultat qui peut faire économiser plus d'argent à l'agent, et le coût d'un autre agent(il profite moins de cette solution), le résultat de la négociation est optimal Pareto.

L'efficacité Pareto mesure les avantages globaux et ne nécessite pas de comparaison des services publics.

Stabilité : Nous avons prédit qu'un agent est en interaction doit être en mesure de discuter les actions possibles des autres agents. La négociation est traitée dans plusieurs domaines comme les guerres, les jeux stratégiques ou football c'est pour cela un agent doit être capable de connaitre la stratégie de l'adversaire ; cette dernière on peut le trouver dans le domaine de théorie de jeux dans le théorème

d'**Equilibre de Nash :**

- Deux stratégies, S1 de l'agent A et S2 de l'agent B, sont dans un équilibre Nash si :

dans le cas où l'agent A adopterait S1 l'agent B ne peut pas faire mieux que d'utiliser S2

dans le cas où l'agent B adopterait S2 l'agent A ne peut pas faire mieux que de d'utiliser S1[36].

Par exemple, dans un match de football si l'un de deux équipes choisi la stratégie d'attaque pour gagner des buts, l'équipe adversaire automatiquement doit adopter la stratégie de défense pour éviter ces buts-là.

La simplicité de calcul : on dit qu'un protocole est simple si les agents peuvent facilement choisir une stratégie parfaite. De plus, un protocole doit être efficace dupoint de vue des calculs nécessaires pour déterminer la stratégie optimale.

5. Stratégie de négociation :

La stratégie de négociation d'un agent est une spécification de la séquence d'actions (généralement offres ou réponses) que l'agent envisage de faire pendant la négociation. Il y aura généralement de nombreuses stratégies compatibles avec un protocole particulier, dont chacune peut produire un résultat différent.Pour étudier la stratégie de négociation on trouve trois approches principales, la théorie des jeux fournit des modèles théoriques abstraits dont les propriétés des accords et des processus ont été formellement prouvées.

Au lieu de cela, l'heuristique fournit Modèles basés sur des hypothèses réalistes, telles que la rationalité limitée, qui sont évalués empiriquement et mis en œuvre dans des applications pratiques. Enfin, la méthode d'argument Les lieux où les agents échangent des offres et des arguments.

La théorie des jeux : Cette stratégie relie les fonctions d'utilité au protocole de concession monotone. En cas de désaccord, la perte de l'agent reconnu est inférieure à la perte de l'agent qui a fait moins de concessions. Le facteur de risque de ZEUTHEN (1930) évalue la propension d'un agent à risquer le désaccord. Il correspond au rapport entre sa perte maximale d'utilité s'il concède et sa perte d'utilité en cas de désaccord [36].

Par conséquent, les négociations commencent avec l'offre préférée. Ensuite, il suffit de négocier ce n'est pas encore fini, l'agent calcule à chaque fois ses facteurs de risque et les facteurs de risque de la contrepartie. Où ses facteurs de risque sont inférieurs ou égaux à ceux de son adversaire, il fait donc des concessions assez minimum. Sinon, il répétera la citation précédente. Grâce à suffisamment d'offres minimales, nous désigne l'offre fera basculer de l'ordre vers la prochaine série de facteurs de risque (en supposant L'adversaire restera ferme) tout en transférant une utilité minimale à l'adversaire.

Ainsi, une négociation entre deux agents qui adoptent cette stratégie consiste en une alternance de concession minimale. On peut remarquer que cette stratégie suppose que l'agent dispose d'une information parfaite, c.-à-d. qu'il connaisse les préférences de son interlocuteur.

Approche heuristique : il s'agit d'une version continue du protocole d'enchère alternatif avec une date limite. Lorsqu'un offre reçu, il va évaluer par la fonctionutilitaire multi-attribut. Les offres sont générées par une combinaison linéaire de fonctions simples (appelées tactiques). Les tactiques calculent les valeurs d'attribut(prix, quantité, qualité, etc.) en fonction de critères. Ainsi, trois familles de tactiques sont suggérées :

Les tactiques temporelles qui déterminent quand un agent doit concéder.

Les tactiques qui contraignent les offres en fonction des ressources disponibles.

Les tactiques imitatives qui déterminent le comportement de l'agent enfonction de celui de son opposant [36].

Selon la fonction de concession d'une tactique de temps, l'agent initie la négociation en proposant une valeur maximale (si l'utilité diminue à mesure que la valeur d'attribut

diminue), puis des concessions Jusqu'à ce que la valeur de rétention soit atteinte avant la date limite (cette fonction est paramétrée). Fonction Les polynôme sont initialement produit plus rapidement que les fonctions exponentielles. Ainsi, on distingue deux tactiques temporelles :

• la tactique ***Boulware*** qui consiste à rester sur ses positions jusqu'à la date butoir poury proposer sa valeur de réserve ;

• la tactique ***Conceder*** qui consiste à concéder rapidement [37].Selon la fonction de concession réciproque, l'agent reproduit proportionnellement le comportement de son opposant. La fonction peut être affinée en pseudo-aléatoire ou par calculez le comportement de l'adversaire dans la fenêtre de temps.

L'approche argumentative : l'argumentation est une approche qui permet aux agents, en plus d'échanger des propositions, d'apporter des justifications concernant les propositions qu'ils, soumettent et leurs décisions (i.e., acceptationou rejet d'une proposition). Chaque agent peut fournir des arguments accompagnantses propositions dans le but d'expliquer les raisons pour lesquelles les autres devraient les accepter. Lorsqu'un agent rejette une proposition, il peut aussi

formuler des critiques expliquant les raisons pour lesquelles cette proposition est inacceptable. Cela permet à certains agents de modifier leurs croyances et préférences. Ces échanges d'informations additionnelles conduisant les agents à s'influencer mutuellement, peuvent aussi avoir un impact important sur le résultat d e l a négociation [37].

En résume, Une stratégie est constituée de trois composants : une règle d'acceptabilité qui décide si une offre est admissible

une stratégie d'offre qui détermine la concession en fonction des préférences, de l'historique et éventuellement d'une valeur de réserve, du temps et du modèle d'opposant (si disponible).

éventuellement une stratégie d'apprentissage ou d'adaptation qui s'appuie sur un modèle d'opposant [35].

II. Négociation multi-issues :

En plus de ce qui était dénommer dans la partie précédente, il y a d'autres formes de négociation moins connues de grand public, ces négociation sont tout aussi nombreuse et variées [34] Parmi ces négociations nous trouve la négociation multi issues.

1. Négociation multi issues

La négociation multi issues comme leur nom indique, sont des négociations qui impliquent des problèmes indépendants les uns des autres .elle sont directement opposée aux enchères qui s'impliquent un seul attribut d'un seul problème. Cette forme de négociation est cependant très rependue et à la base de nombreuse variantes de négociation.

2. Modèle de négociation multi issues :

Nous prenons le modèle de la négociation multi issues suivant ; Supposons que l'acheteur, **b**, et le vendeur, **S**, qui ont des délais inégaux, négocient le prix de deux biens / services distincts, **X** et **Y**. Ici ; **Ta** dénote la date limite de l'agent, **a**, pour parvenir à un accord sur les deux issues.

La négociation sur toutes les issues doit être achevée avant la première des deux date limites. Nous considérons deux biens / services afin de simplifier la discussion,mais ceci est un cadre général qui travaille pour plus de deux biens / services.

Etat d'information des agents :

En dénote les valeurs de réservation de l'acheteur pour X et Y par RP_x ,RP_Y et l'autre du vendeur sont respectivement RP_x RPY aussi S_x dénote la stratégie de l'agent pour l'issue X et SY indique la stratégie de l'agent a pour l'issue Y. alors l'état d'information de l'acheteur est :

$$I^s=\{RP_x^s, RP_y^s, T^s, s, S_x^b, S_y^b, L_t^b, L_x^b, L_y^b\}$$

Lt, Lx ,Ly sont trois distributions de probabilité qui dénote ses croyances sur les paramètres de l'adversaire ; une autre fois RPX;RP_Y et S_x dénote les croyances de l'acheteur concernant le délais de vendeur, sa valeur de réservation pour X et sa valeur de réservation pour Y respectivement.De manière similaire, l'état des informations du vendeur est défini comme :

$$I^s=\{RP_x^s, RP_y^s, T^s, s, S_x^b, S_y^b, L_t^b, L_x^b, L_y^b\}$$

L'état d'information de chaque agent est sa connaissance privée.

Protocole de négociation :

Nous utilisons un protocole de négociation d'offres alternatives. L'un des agents commence par faire une offre combinée. Un autre agent peut accepter ou rejeter une partie

de l'offre (un seul issue) ou l'offre complète. S'il rejette l'offre complète, il envoie une contre-offre combinée. Le processus de proposition d'une fusion se poursuit jusqu'à ce qu'un accord soit conclu sur l'un des points. Après L'agent ne fait qu'une offre sur les issues restantes (c'est-à-dire qu'une fois qu'un accord est conclu sur l'issue, il ne peut être renégocié). Lorsqu'un accord est conclu sur deux

questions ou que le délai est atteint, les négociations sont conclues. Soit $S_{0x}(t)$ le prix généré par le meilleur agent de l'agent, b, Stratégie d'émission X au temps t.Par conséquent, l'action telle que prise par l'agent, s, au moment t d'une offre unique.Son action sur une offre combinée, est définis comme:

$$A^s(t, X^t_{b\to s}, Y^t_{b\to s}) = \begin{cases} \text{Quitter} & \text{si } t > T^s; \\ \text{Accepter } X^t_{b\to s} & \text{si } X^t_{b\to s} \geq S^b_{0x}(t), \\ \text{Accepter } Y^t_{b\to s} & \text{si } Y^t_{b\to s} \geq S^b_{0y}(t), \\ \text{Offre } S^s_{0x}(t') \text{ à } t' & \text{si } X^t_{b\to s} \text{ n'est pas accepté,} \\ \text{Offre } S^s_{0y}(t') \text{ à } t' & \text{si } Y^t_{b\to s} \text{ n'est pas accepté.} \end{cases}$$

La fonction d'utilité des agents est comme suit :

$$U^a(P_x, P_y, t) = \begin{cases} (RP^b_x - P_x)(\delta^b_X)^t + (RP^b_y - P_y)(\delta^b_y)^t & \text{Pour b} \\ (P_x - RP^s_x)(\delta^s_X)^t + (P_y - RP^s_y)(\delta^s_y)^t & \text{Pour s} \end{cases}$$ [37]

Conclusion

Les négociations jouent un rôle vital dans le cas où les agents doivent résoudre des conflits pouvant mettre en danger les comportements coopératifs. Dans ce chapitre nous avons présenté les différents concepts de négociation qui permettent à comprendre parfaitement la négociation entre les agents. Nous pouvons dire que par la négociation, nous permet de résoudre les conflits sans touchez l'aspect autonomie de l'agent car ce dernier peut continuer ou s'arrêter la négociation.

Contribution

Chapitre 03 : Négociation par renforcement basée agent

Introduction :

Dans ce chapitre, nous allons bâtir un agent négociateur basé sur la technique d'apprentissage par renforcement, cette technique est soucieuse de ce que fera l'adversaire cette dernière est plus difficile que découvrir ce qu'il veut.

Dans ce chapitre ; nous présentons une bref description de notre agent, ensuite nous allons voir l'approche d'apprentissage par renforcement ; finalement, nous proposons une conception d'un agent négociateur, ainsi que leur stratégie d'apprentissage, d'acceptation /refus, d'offre et de concession.

1. Description de notre agent :

Dans ce mémoire, nous nous intéressons particulièrement à la modélisation d'un agent capable de négocier un agent adversaire et de maximiser sa utilité pour gagner le tour de négociation à l'aide d' une stratégie d'apprentissage par l'apprentissage par renforcement qui lui permet de connaitre ce que fera l'adversaire et lui proposera des offres plus utilitaires chez nous et satisfaisante chez l'adversaire. Notre agent doit vérifier les propriétés suivantes :

La réactivité : l'agent doit être capable à répondre aux évènements extérieurs.

L'autonomie : il doit être capable d'agir sans l'intervention d'un tiers.

La rationalité : l'agent être capable d'effectuer l'action qui lui permet de maximiser leur performance à la base de ses connaissances précédentes et ses séquences perspectives.

La possession d'un ou plusieurs objets.

La situation, l'agent peut percevoir son environnement à travers ses captures et peut agir lerésultat de son effectuer.

Le tempérament égoïste : l'agent privilégiera à tout prix son propre intérêt [31].

2. L'apprentissage profond :

L'apprentissage profond est un sous type de l'apprentissage automatique qui vise à former et à enseigner la machine à exécuter des fonctions humaines telles que la distinction d'objets visuels et l'identification du son et de l'image. Au lieu d'organiser les données, l'apprentissage profond définit ses propres paramètres de base qui permettent à la machine d'apprendre de manière indépendante.

Le *deep learning*, ou apprentissage profond, appartient à la grande famille de l'intelligence artificielle. Plus précisément, il constitue un sous-ensemble de la *machine learning*, et fait appel à des types particuliers de réseaux de neurones artificiels. Il présente donc des caractéristiques similaires à ces techniques, notamment la capacité d'apprentissage de façon autonome. Pour cela on a trois approches d'apprentissage selon le cas d'utilisation :

Apprentissage supervisé.

Apprentissage non supervisé.

Apprentissage par renforcement.

3. Apprentissage par renforcement :

L'apprentissage par renforcement est un domaine de l'apprentissage automatique. Il s'agit de prendre des mesures appropriées pour maximiser les bénéfices dans une situation donnée. Il est utilisé par divers logiciels et machines pour trouver le meilleur comportement possible ou le meilleur chemin à suivre dans une situation donnée. L'apprentissage par renforcement diffère de l'apprentissage supervisé en ce sens que, dans l'apprentissage supervisé, les données de formation ont la clé de réponse ; le modèle est donc formé avec la bonne réponse alors que dans l'apprentissage par renforcement, il n'y a pas de réponse, mais l'agent de renforcement décide quoi faire pour effectuer la tâche donnée. En l'absence de jeu de données de formation, il est tenu d'apprendre de son expérience [38].

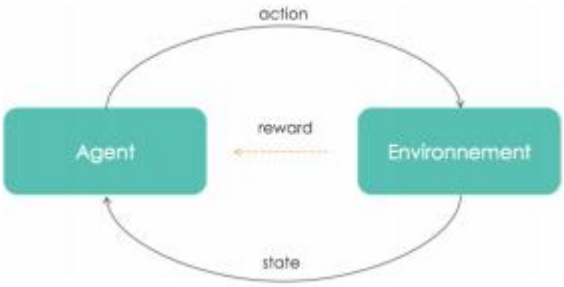


Figure 7. Exemple d'un agent utilise l'apprentissage par renforcement [38].

Presque tous les problèmes du RL peut être formalisées à l'aide du processus de décision de Markov (MDP), est un processus sans mémoire avec une séquence d'états aléatoire S_1… …Si

qui utilise la propriété de Markov qu'il dit que « **la nouvelle action et l'état de l'agent ne dépend pas du passé actions ou états** », ce dernier se compose d'un tuple de quatre éléments (S, A, P, R) ; où :

S : l'ensemble d'états finis.

A : l'ensemble des actions finis.

P : la probabilité d'une transition d'état S s à s` lorsque l'action A a est choisie :

$$P = (s`|s,a) = P(s_{t+1} = s|s_t = s;\ a_t = a).$$

R : la récompense reçu après la transition de l'état s à l'état s` à cause d'une action a.

La figure 7 se présente ce processus, un agent interagie avec l'environnement et effectuer des actions, à chaque action l'environnement réagit et génère un nouvel état.

4. Le champ des termes du RL dans notre modèle :

Agent ( ) : une entité qui peut percevoir notre agent négociateur.

Environnement ( ) : si où notre agent est présent ou tourné la plateforme de simulation de denégociation GENIUS.

Action ( ) : est les actions effectuées par notre agent négociateur au cours de négociation accepte/ refus, contre-offre, concession.

State ( ) : la situation retourné par l'environnement après chaque action de notre agent gagner le tour / perdu le tour.

Reward ( ) : un retour d'information renvoyé à l'agent depuis l'environnement pour évaluer l'action de l'agent et toujours prend la valeur 0 sauf dans le cas où Reward= utilité du final accord r =1/ avec r ϵ [0,1].

Policy ( ) : la stratégie appliqué pour la prochaine action en fonction de l'état actuel de notre agent.

Valeur ( ) : attendu à long terme avec le facteur d'actualisation et opposée à la récompense à courte terme.

Q-valeur ( ) : similaire à la valeur, mais elle prend un paramètre supplémentaire comme action courante a. Les algorithmes d'apprentissage par renforcement sont utilisés principalement dans le domaine des applications d'IA et les applications de jeu. Dans notre

mémoire et pour traiter le problème d'elicitations des préférences dans une négociation multi issues non linéaire nous choisissons d'appliquer l'algorithme SARSA.

5-Explication du SARSA dans notre modèle

En intelligence artificielle, plus précisément en apprentissage par renforcement, **SARSA** est un algorithme d'apprentissage. Son nom est l'acronyme de *State-Action-Reward-State-Action* (Etat-Action-Récompense-Etat-Action)1. C'est un algorithme on-policy : il utilise la politique en train d'être apprise pour mettre à jour les valeurs internes apprises.

Le nom SARSA signifie Etat-Action-Récompense-Etat-Action qui est la suite des éléments mathématiques considérés par l'algorithme :

- l'algorithme considère l'état courant s (par exemple, la position d'un robot dans un environnement et la position de ses bras)
- puis il choisit une action à exécuter en fonction de ce qu'il a déjà appris, mais aussi un biais d'exploration pour éveiller sa curiosité et essayer des actions non préconisées. Il exécute cette action
- il reçoit alors une récompense r. Par exemple, si le robot est toujours en vie, on peut décider de lui une récompense de 1€, alors que s'il tombe, il perd 100€ (i.e. une récompense négative de -100€)
- il perçoit le nouvel état s' (par exemple, sa nouvelle position)
- il choisit la nouvelle action a' exécuter

La mise à jour de ses valeurs internes tient compte de valeur Q(s', a') et Q(s, a) et la récompense r. On dit que SARSA est un algorithme on-policy car la mise à jour tient compte en particulier de Q (s', a'), où a' a été choisi par la politique en cours d'apprentissage.

Pseudo-code

Voici le pseudo-code de l'algorithme SARSA. Q(s, a) désigne la valeur sur le long terme estimée d'exécuter a dans s. Le choix des actions est réalisée à partir des valeurs Q(s, a).

```
initialiser Q[s, a] pour tout état s, pour toute action a de façon arbitraire, mais Q(état terminal, a) = 0 pour tout action a

répéter
    //début d'un épisode
```

```
    initialiser l'état s
    choisir une action a depuis s en utilisant la politique spécifiée par Q (par exemple ε-greedy)



    répéter
        //étape d'un épisode
        exécuter l'action a
        observer la récompense r et l'état s'
        choisir une action a' depuis s' en utilisant la politique spécifiée par Q (par exemple ε-greedy)
Q[s, a] := Q[s, a] + α[r + γQ(s', a') - Q(s, a)]

s := s'
        a := a'
    jusqu'à ce que s soit l'état terminal
```

Proprieté

- L'algorithme SARSA est on-policy : la politique apprise est la même que celle utilisée pour faire les choix des actions à exécuter. Un algorithme proche est le Q-learning, mais qui off-policy.
- L'algorithme de type TD : on utilise l'état suivant s' pour mettre à jour la valeur en l'état courant s.(39)

6- La conception de notre agent négociateur :

On expérimentera sur la plate-forme GENIUS, consacré à simuler les négociations automatique entre des agents intelligents le problème posé dans ce mémoire est d'élaborer un agent capable d'effectuer une élicitation des préférences au cours d'une négociation multi issues non linéaire.

L'espace de résultats dans une négociation est le produit cartésien des issues, représenté par : $\Omega = \{W1 \dots WN\}$ où wi représente les résultats possibles et N est le nombre total de

résultats possibles. Chaque agent relie une utilité à un résultat w, en fonction de son profil de préférence, noté par UA(w) ϵ [0, 1] pour l'agent A, l'espace de résultat définie un scénario de négociation avec le profil de préférence.

Chaque agent a une valeur réservée qui est l'utilité minimale accepté par l'agent, il n'acceptera jamais qui ne correspond pas à cette valeur, nous la dénotons par B. Il n'est pas facile d'élaborer un cadre d'apprentissage commun pour le renforcement des négociations multi-issues. Un défi majeur est de permettre à un espace d'état et d'action commun d'être représenté de manière compacte et simultanément contesté arbitrairement par le concepteur d'agent pour s'adapter à différents paramètres de négociation.

Et pour faciliter ce défi de l'état et l'action, nous doit impliquons trois composants principaux dans la stratégie de l'agent négociateur : une stratégie de proposition pour décider quoi offrir, un modèle d'adversaire pour estimer l'utilité par différentes offres, et une stratégie d'acceptation qu'aide à prendre une décision d'accepter l'offre de l'adversaire ou proposé la prochaine proposition. L'astuce ici est d'associe la stratégie de négociation en une stratégie de proposition, un modèle d'adversaire et une stratégie d'acceptation.

Afin de fournir une négociation par une interface d'apprentissage par renforcement, il faut que nous décrivons les états .et les actions qui favorisent l'interaction avec l'environnement d'une manière indépendante du domaine. Ceci rendu difficile par deux caractéristiques de Ω :

1- Sa taille est exponentielle par le nombre des issues.

2- L'espace de résultat doit être unique pour chaque combinaison des issues [40].

Notre stratégie de négociation est constituée d'une stratégie d'apprentissage, une stratégie d'acceptation, une stratégie de proposition et une stratégie de concession sont présentés ci-dessous dans l'organigramme suivant :

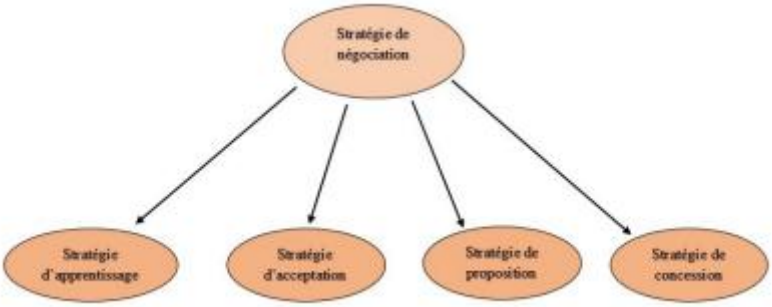


Figure 8. Les composants de la stratégie de négociation. [21]

Nous allons détaillées chaque une des stratégies précédente dans le paragraphe suivant :

6.1. Stratégie d'apprentissage :

Tout d′abord, nous fixons la valeur **0.94** comme cible pour l´algorithme. l´algorithme essaie d´atteindre cette valeur, l´algorithme commence par donner des valeurs initiales pour la valeur de **Bid** ainsi que **utility**, puis l´algorithme commence à chercher un nouveau **Bid** pour qu'il soit meilleur que le précédent **Bid**, si ce dernier est meilleur que le précédent, alors l´algorithme cache sa valeur et la considère comme la meilleure action qui peut être présentée.

L'agent doit toujours sélectionner la commande à exécuter. Il est donc nécessaire faites un compromis entre exploitation et exploration. Dans notre cas, nous avons choisi le type fonction de décision ε- Greedy. Ce principe conserve le principe du choix d'un comportement gourmand dans la plupart des cas.

Pour obtenir un équilibre entre l'exploitation et l'exploration, nous utilisons ce que nous appelons la stratégie Epsilon Greedy. Avec cette stratégie, nous définissons un taux d'exploration ε que nous fixons au départ à 1 . Ce taux d'exploration est la probabilité que notre agent explore l'environnement plutôt que de l'exploiter. Avec ε = 1 , il est certain à 100 % que l'agent commencera par explorer l'environnement. Au fur et à mesure que l'agent en apprend davantage sur l'environnement, ε se dégradera à un taux que nous fixons de sorte que la probabilité d'exploration devient de moins en moins probable à mesure que l'agent en apprend de plus en plus sur l'environnement. L'agent exploitera davantage l'environnement une fois qu'il aura eu l'occasion de l'explorer et d'en apprendre davantage sur lui. Pour déterminer si l'agent choisira l'exploration ou l'exploitation à chaque étape, nous générons un nombre aléatoire entre 0 et 1 . Si ce nombre est supérieur à ε , alors l'agent choisira sa prochaine action via l'exploitation. Sinon, son action suivante sera choisie via l'exploration.[21]

6.2. La stratégie de proposition :

Pour la stratégie de proposition, notre agent va apprenait que veut l'adversaire et lui proposé une offre, premièrement nous allons déclarer une valeur **d= 0.94** qui représente l'utilité objective, en deuxième place il faut explorer l'environnement pour trouver la meilleur valeur qui est proche de d si l'algorithme trouver un B**id** qui est meilleur que le dernierB**id** l'algorithme stocker ce B**id** dans la mémoire pour proposer ce B**id** comme un offre. En utilisant **e-Greedy** pour faites un compromis entre exploitation et exploration. Si

p < 0.6 on a générer nouveau B**id** et comparer **l'utility** de ce **Bid** avec la meilleur B**id** que on a trouver si **l'utility** de ce **Bid** est meilleur que **l'utility** de max B**id**, l'algorithme stocker ce **Bid** comme la meilleur et proposer l'offre, si **p> 0.6** l'algorithme proposer la meilleur B**id** direct sans faire l'exploration.

6.3. La stratégie d'acceptation :

La stratégie d'acceptation est basée sur la combinaison de condition de temps (t> T) et utilité(Ua), elle est représentée comme suit :

Tant que t > T faire :
  Si $U_A(W_{B\to A}^t)$ > d alors
    → A := accepte l'offre de l'opposant ;
  Sinon si $[\alpha\, U_A(W_{B\to A}^t) > U_A(W_{A\to B}^{t+1})]$
  Si non A := refus l'offre de l'opposant ;

[21]

6.4. La stratégie de concession :

Nous proposons une condition pour effectuer une concession selon le temps, ainsi quand il reste 10% de deadline l'agent commence à faire des concessions de la façon suivante : il diminue lavaleur de l'utilité objective jusqu'elle égale la valeur de l'utilité réservée

Tant que d > β et t<= 10%T faire
  d := $d - 10\%\, d$;

[21]

Nous résume chaque stratégie parmi les stratégies précédentes par les organigrammes suivantes :

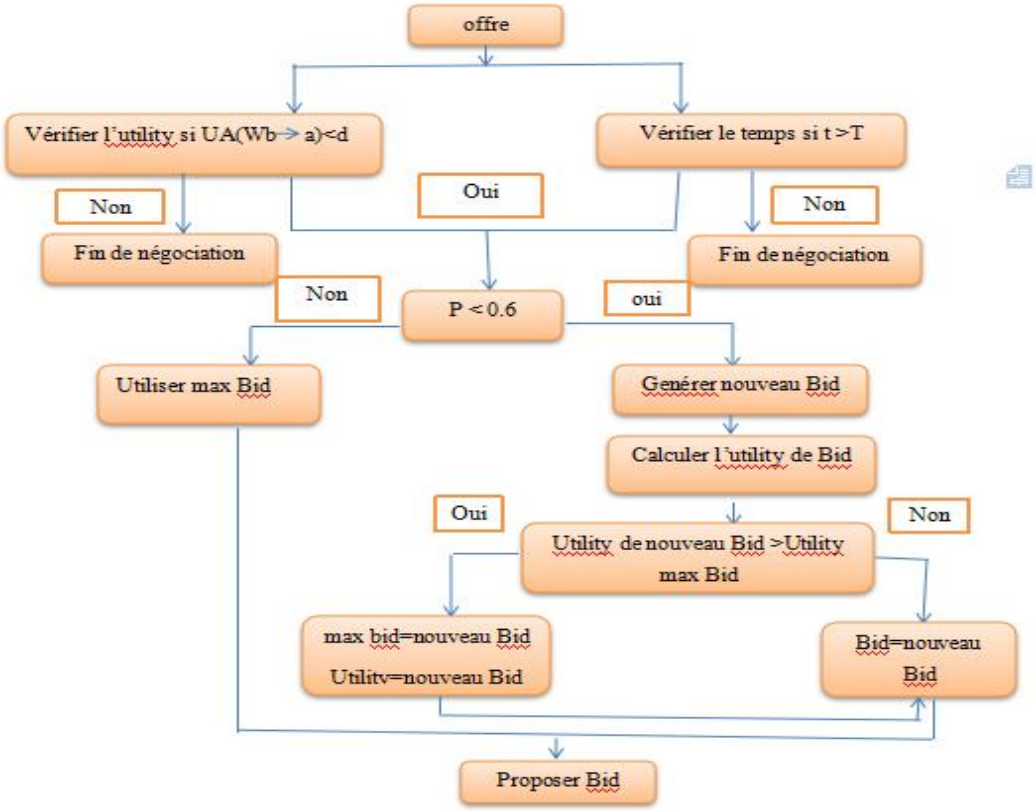


Figure 9. La stratégie de proposition.

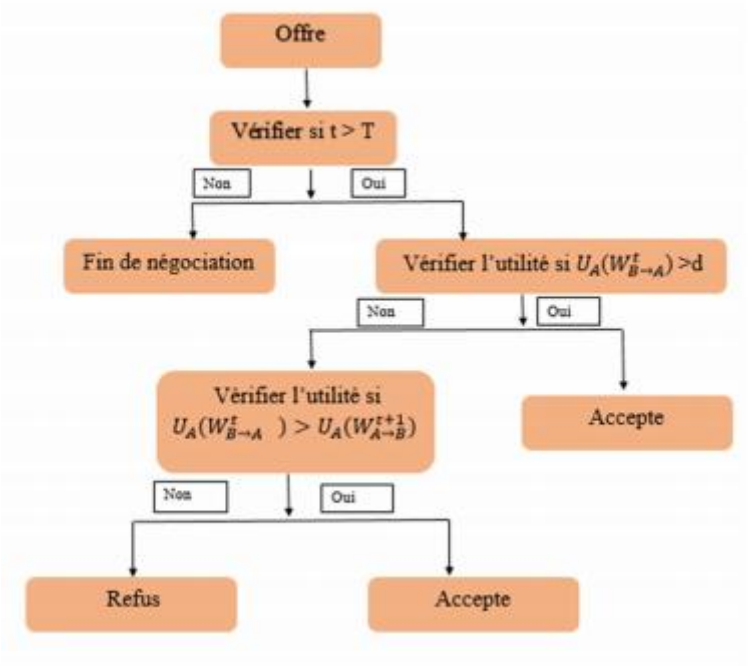


Figure 10. La stratégie d'acceptation [21]

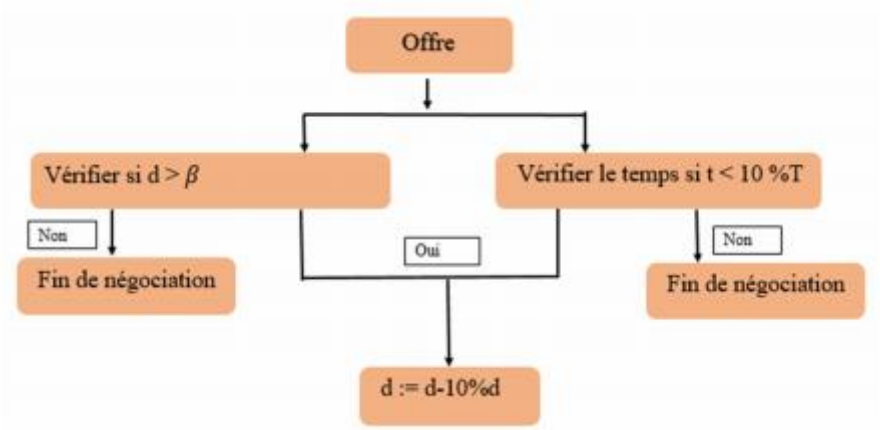


Figure 11. La stratégie de concession [21]

Conclusion :

Dans ce chapitre nous avons établi les principes de notre modèle de conception d'agent négociateur à travers l'algorithme SARSA, et défini également l'apprentissage par renforcement ainsi que la confirmation de choix d'algorithme SARSA.

L'objectif de chapitre suivant est de valider notre modèle de conception dans la plateforme de simulation de la négociation automatique GENIUS.

Chapitre 04 :

Implémentation et résultat

Introduction

Dans le chapitre précédent nous avons présenté le principe de notre modèle de conception d'un agent négociateur. Ce chapitre 4 est consacré à l'étude comportementale de cet agent sur négociation multi-issues non linéaire dans la plateforme GENIUS.

Dans ce chapitre, nous appliquerons l'un des algorithmes d'apprentissages par renforcement pour la négociation multi-issues non linéaire sous la forme de test avec différents agents compatibles sous la plateforme utilisé ; ainsi que nous présenté les logiciels utilisés.

1. La plateforme GENIUS :

L'environnement GENIUS (**G**eniric **E**nvironment for **N**egotiation with **I**ntelligent multi-purpose **U**sage **S**imulation) est Conçu pour promouvoir des stratégies de négociation. Avec GENIUS, les programmeurs peuvent se concentrer sur la conception de stratégies. À cette fin, GENIUS fournit un environnement flexible et facile à utiliser pour la mise en œuvre d'agents et des mécanismes de support pour la conception et l'analyse de stratégies d'agents. De plus, le cœur de GENIUS peut être intégré dans un système de support de transaction plus large qui peut prendre en charge

l'intégralité de la négociation du début à la fin. GENIUS se concentre sur les négociations bilatérales, c'est-à-dire les négociations entre deux parties ou agents A et B. Les agents négocient sur les issues qui entrent dans le champ de la négociation, et chaque issue à un éventail d'alternatives ou de valeurs. Les résultats des négociations comprennent chaque issue est mappée sur une valeur et tous les ensembles Ω possibles est appelé l'espace des résultats. La partie négociatrice connaît le domaine de résultat et reste inchangée pour une seule session de négociation.

Le résultat est appelé l'espace des résultats. La partie négociatrice connaît le domaine de résultat et reste inchangée pour une seule session de négociation. Nous supposons en outre que les deux parties ont certaines préférences spécifiées dans le profil de préférence Ω. Ces préférences peuvent être modélisées à l'aide de la fonction d'utilité normalisée U, qui mappe le résultat possible $\omega \epsilon \Omega$ aux nombres réels dans la plage [0, 1]. Contrairement à l'espace de résultats, le profil de préférence de l'agent est une information privée. [41]

1.1. **Présentation des interfaces du GENIUS :**

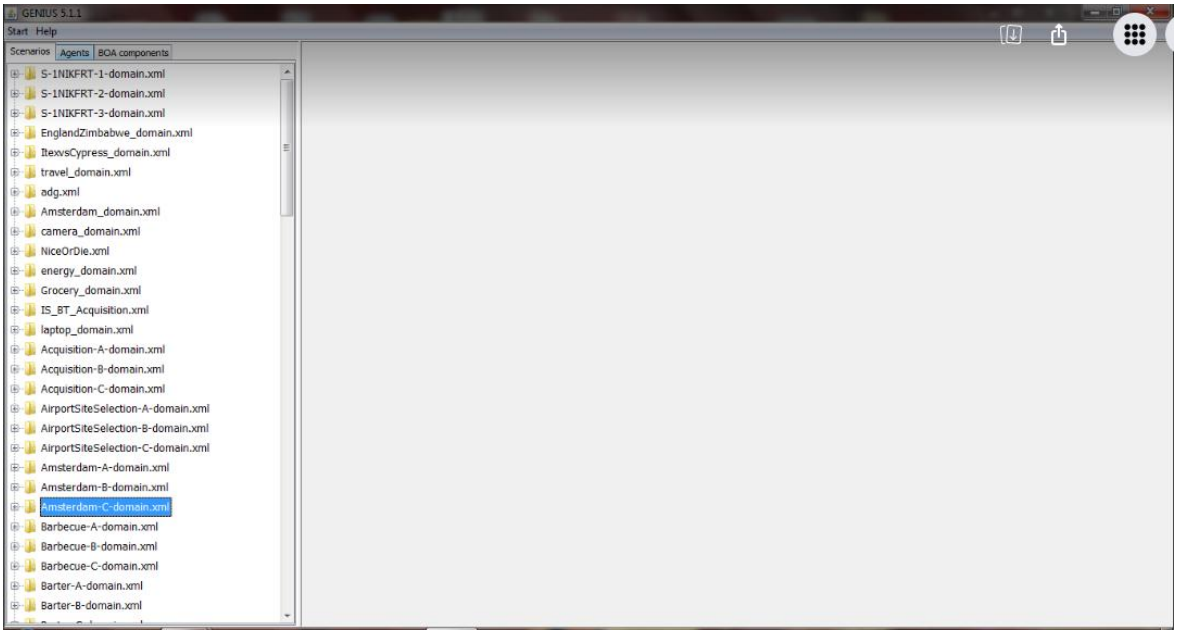


Figure 12. GENIUS juste après le démarrage, à la gauche on a le panneau de composants à la droite Le panneau d'état.

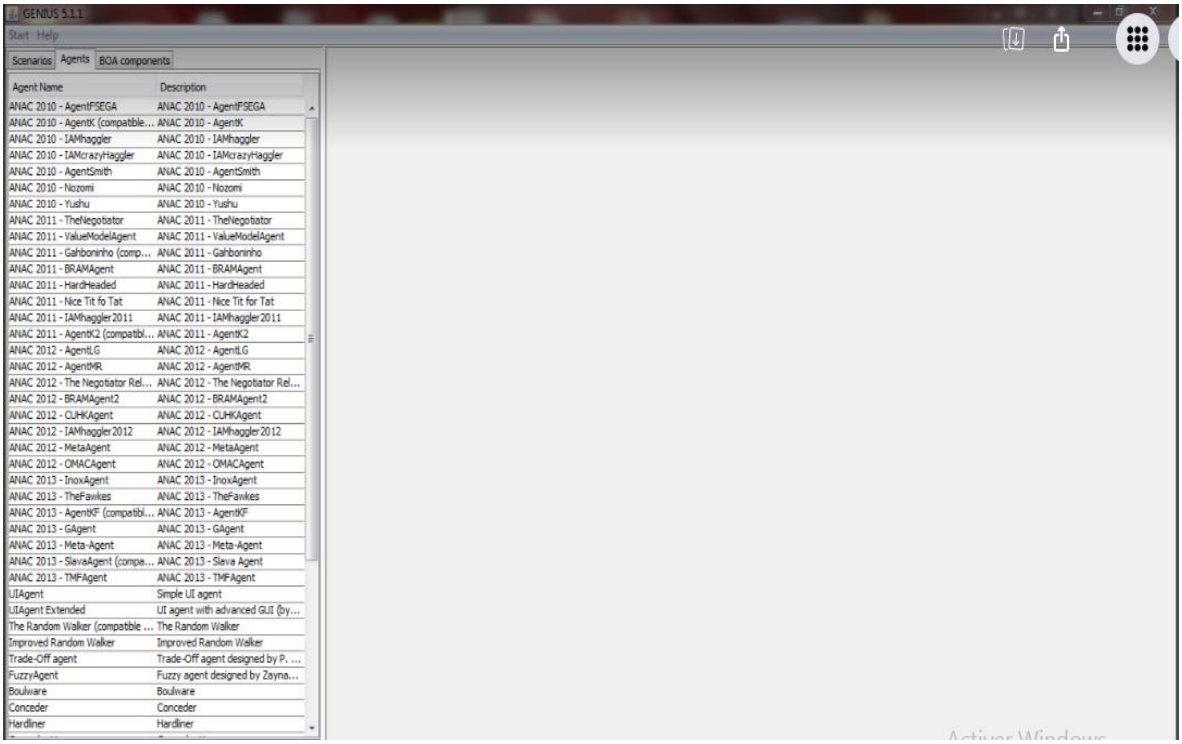


Figure 13. Les agents intégrés avec GENIUS.

.

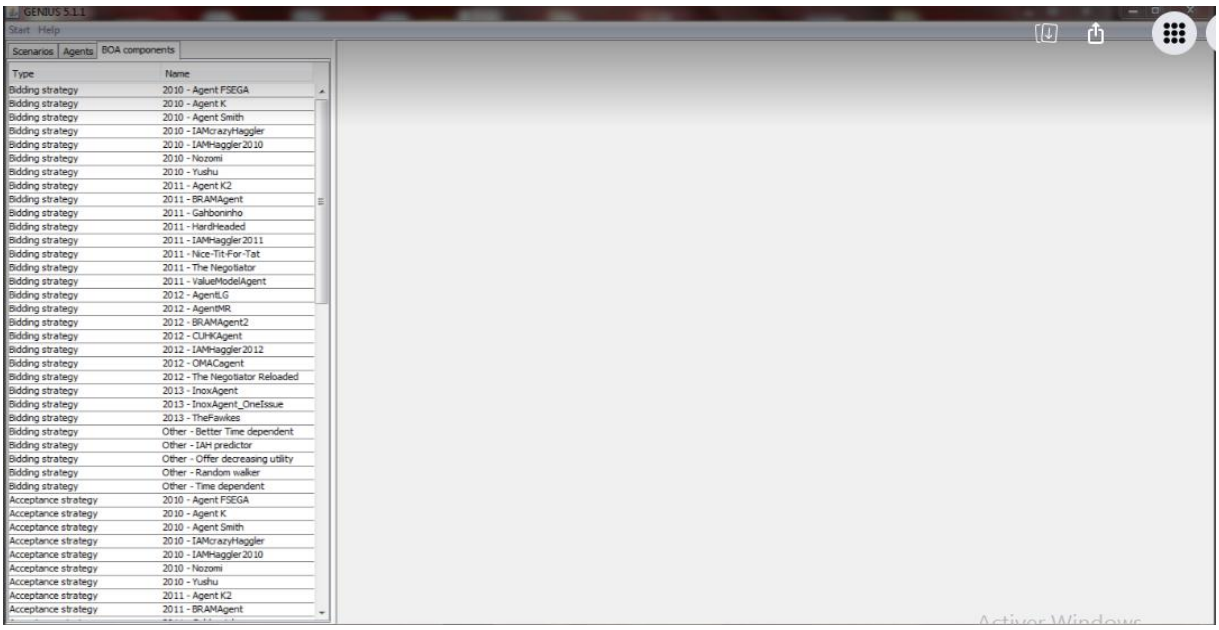


Figure 14. composants BOA ; offrant à l'utilisateur la possibilité de créer et d'appliquer des composants nouvellement développés utiliser une interface utilisateur graphique.

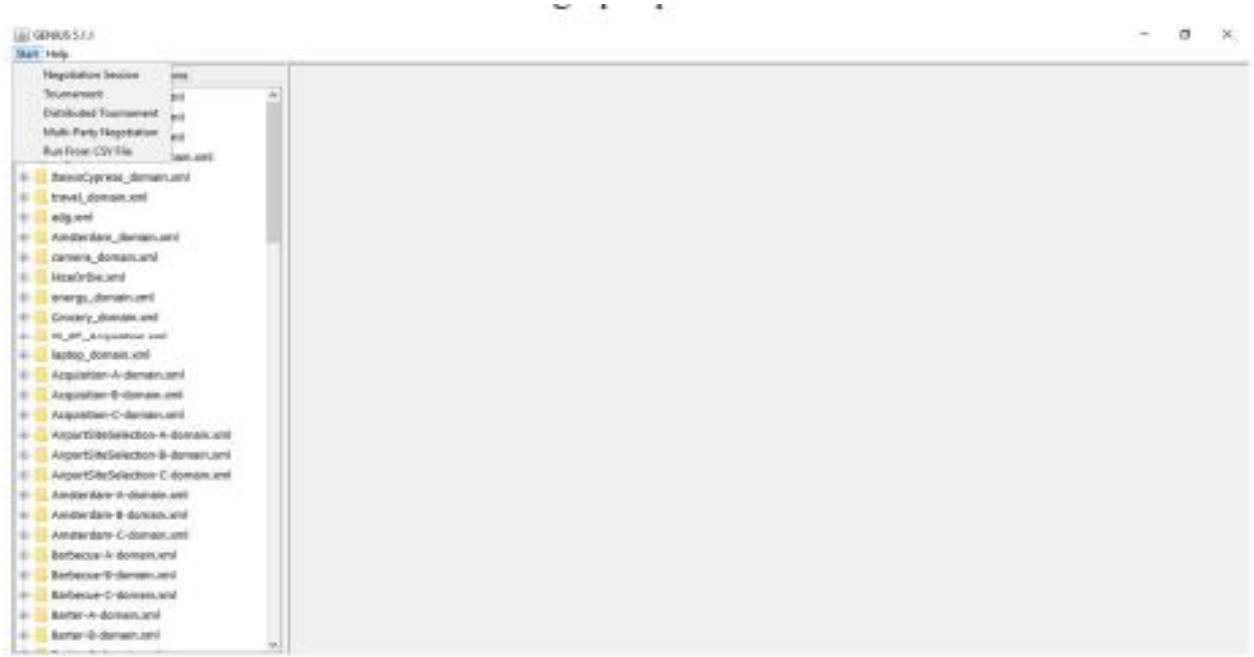


Figure 15. Création d'une négociation session simple.

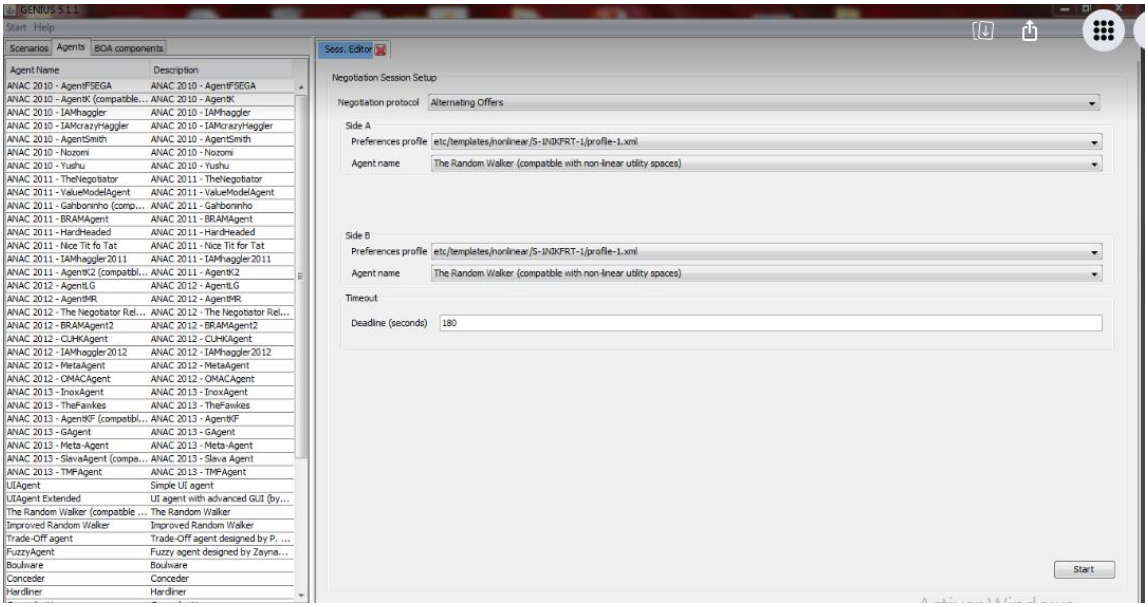


Figure 16. Choisir les paramètres d'une négociation, ainsi que le protocole de négociation, de adline et les agents et leurs préférences, enfin, cliqué « start » pour commencer la négociation.

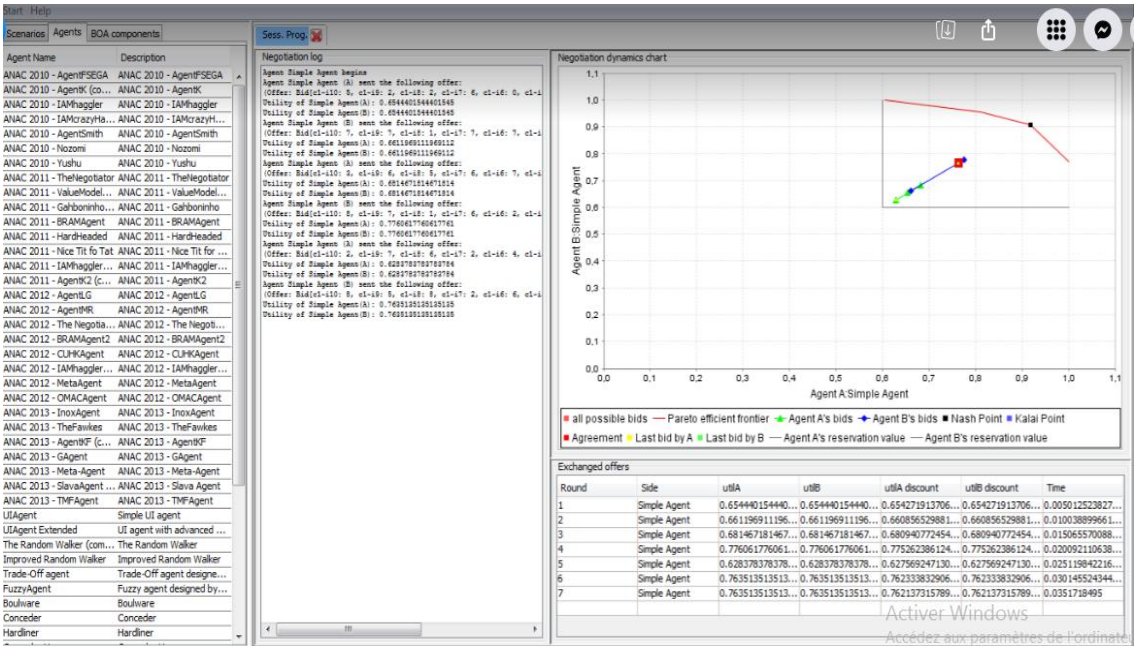


Figure 17. Le résultat de la négociation.

2. La compétition des agents de négociation automatisés

ANAC :

Le Concours international des agents de négociation automatisés (ANAC) est un événement annuel, organisé conjointement avec la Conférence conjointe internationalesur les agents autonomes et les systèmes multi-agents (AAMAS), ou la Conférence conjointe internationale sur l'intelligence artificielle (IJCAI). Le concours de l'ANAC rassemble des chercheurs de la communauté de la négociation et fournit des repères uniques pour évaluer des stratégies de négociation pratiques dans des domaines à enjeux multiples. Les concours ont donné naissance à de nouvelles recherches en IA dans le domaine de la conception d'agents autonomes, accessibles à la communauté de recherche au sens large.

L'ANAC met les chercheurs au défi de développer des négociateurs automatisés efficaces pour les scénarios où les informations sur l'adversaire sont incomplètes. Avec ce concours, la communauté de la recherche dans la négociation automatisée oriente la recherche de ses chercheurs, encourage la conception d'agents de négociation génériques capables de fonctionner dans une variété de scénarios et fournit des repères de performance [42].

Le premier concours a été organisé conjointement avec la neuvième conférence internationale sur les agents autonomes et les systèmes multi-agents (AAMAS-10) et comprenait sept équipe.

3. Eclipse ET java (JDK) :

Eclipse est un environnement de développement (IDE) historiquement destiné au langage Java, même si grâce à un système de plugins il peut également être utilisé avec d'autres langages de programmation, dont le C/C++ et le PHP. Eclipse nous permet de programmer même des agents à partir un lien de connexe entre Genius et Eclipse.

Eclipse nécessite une machine virtuelle Java (JRE) pour fonctionner. Mais pour compiler du code Java, un kit de développement (JDK) est indispensable. [43]

Le kit de développement Java (JDK) est un environnement de développement logiciel utilisé pour développer des applications et des applets Java. Il comprend l'environnement d'exécution Java (JRE), un interpréteur / chargeur (java), un compilateur (javac), un archiveur (jar), un générateur de documentation (javadoc) et d'autres outils nécessaires au développement Java. [44]

4. Discussion des résultants :

4.1. Exemple d'application dans une session de négociation :

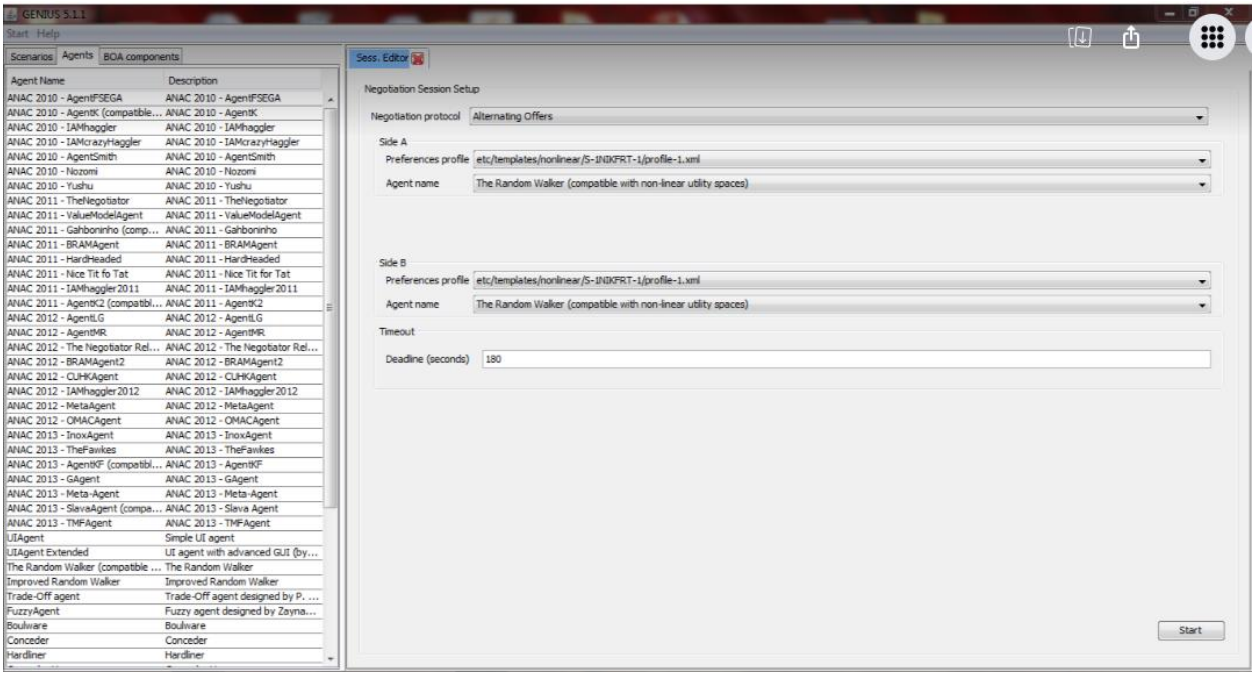


Figure 18. La session de négociation est selon le protocole alternative, notre agent est de la coté « A » nommé« RL agent » avec l'agent KF

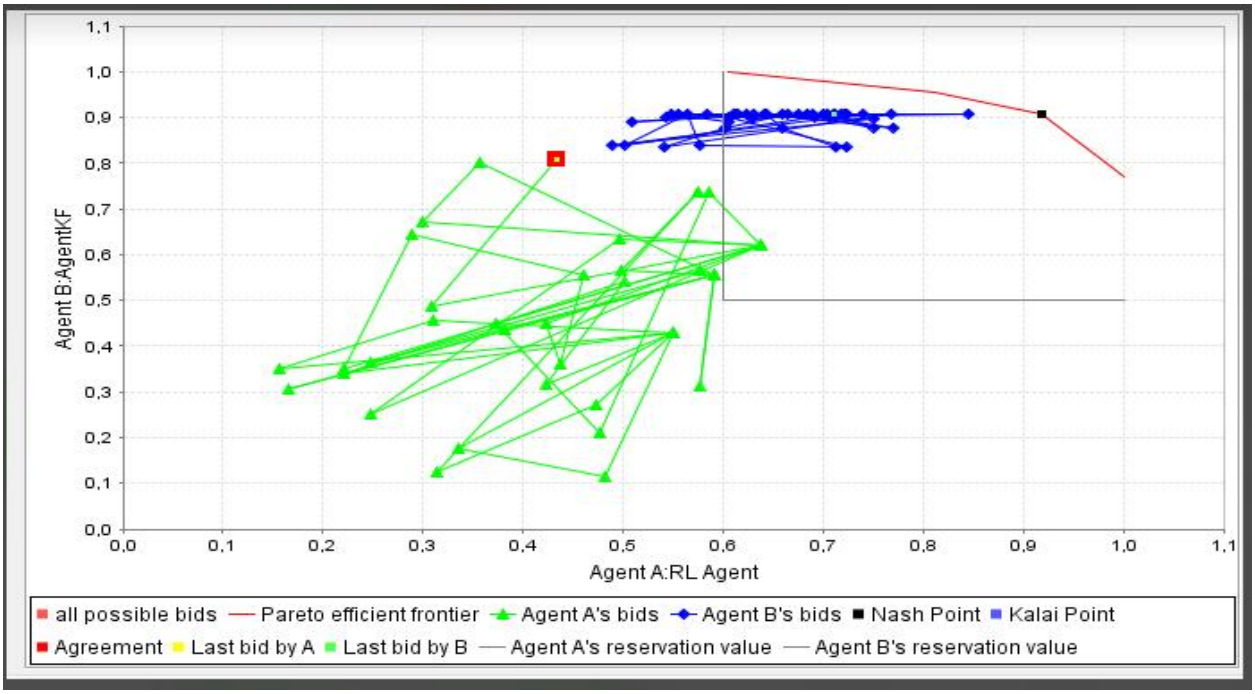


Figure 19. Les résultats selon la négociation session précédente.

Nous remarquons que notre agent est gagné la session avec une utilité max environ 0.917.

Les offres de notre agent au cours la négociation est comme suit :

Agent RL agent (A) begins

Agent (A) sent the following offer:

(Offer: Bid [c1-i10: 7, c1-i9: 7, c1-i8: 3, c1-i7: 6, c1-i6: 8, c1-i5: 9, c1-i4: 1, c1-i3: 2, c1-i2: 6, c1-i1: 9,])

Utility of RL agent(A): 0.5501930501930502

Utility of AgentKF (B): 0.4275568181818182

Agent AgentKF (B) sent the following offer:

(Offer: Bid [c1-i10: 5, c1-i9: 6, c1-i8: 0, c1-i7: 8, c1-i6: 6, c1-i5: 6, c1-i4: 1, c1-i3: 6, c1-i2: 8, c1-i1: 7,])

Utility of RL agent (A): 0.7123552123552124

Utility of AgentKF (B): 0.8338068181818182

……..

Agent RL agent (A) sent the following offer:

(Offer: Bid [c1-i10: 3, c1-i9: 0, c1-i8: 7, c1-i7: 7, c1-i6: 8, c1-i5: 8, c1-i4: 9, c1-i3: 0, c1-i2: 9, c1-i1: 6,])

Utility of RL agent (A): 0.4333976833976834

Utility of AgentKF (B): 0.8082386363636364

Utility of RL agent(A): 0.5501930501930502

Utility of AgentKF (B): 0.8082386363636364

4.2. Résultats obtenus :

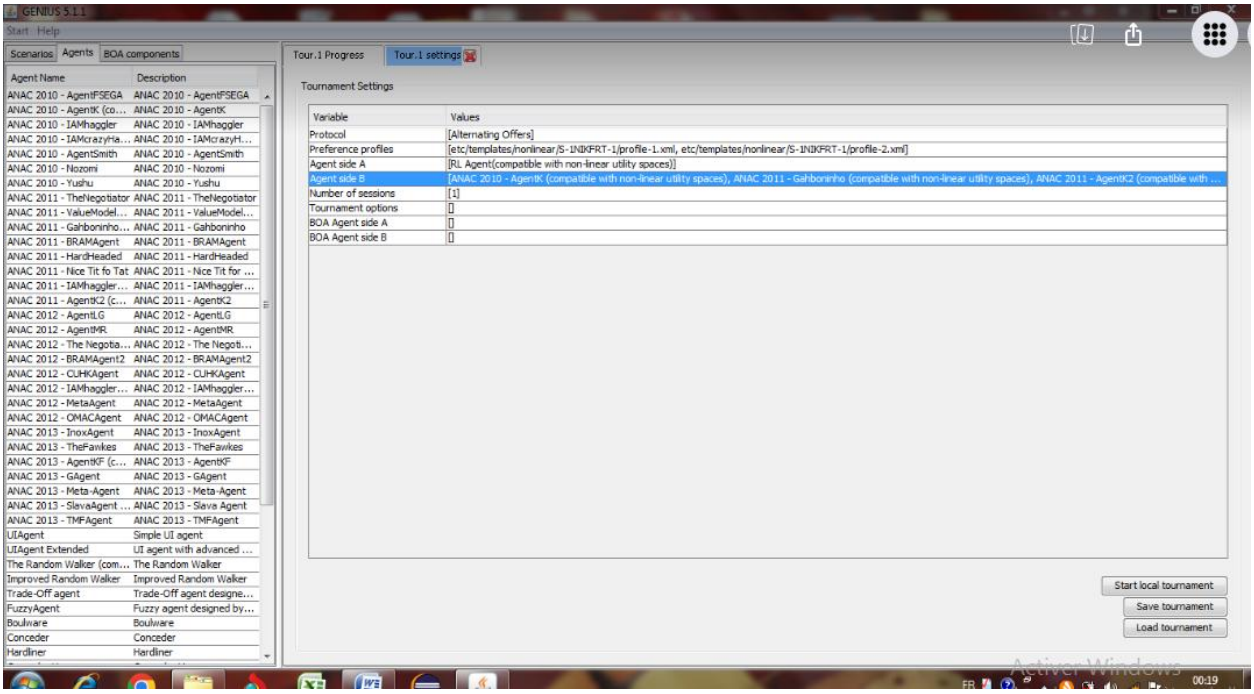


Figure 20. Une session tournoi avec tous les agents compatibles avec les utilités non linéaires sous GENIUS.

Après avoir cliqué sur « start local tournament », Le panneau est ensuite échangé pour une progression du tournoi.

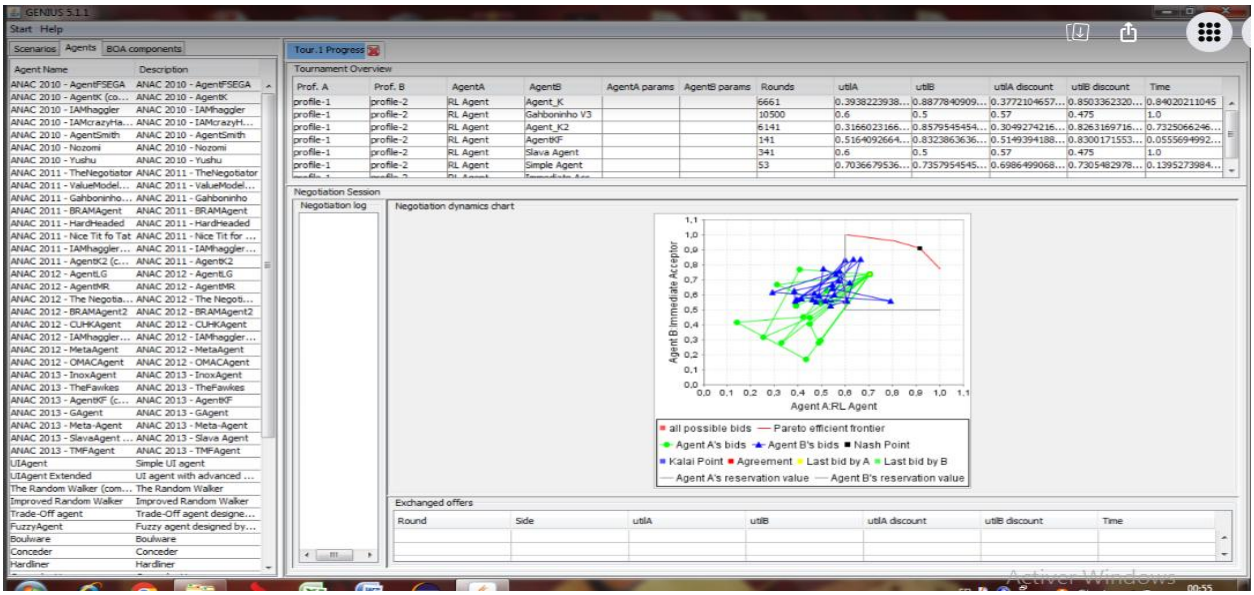
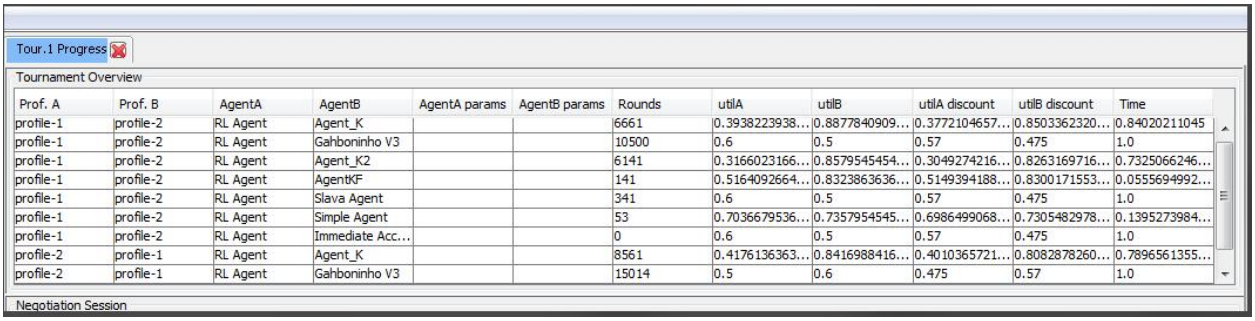


Figure 21. Les résultats du tournoi ; La boîte à outils donne de précieux graphiques Informations pendant les sessions de négociation, y compris : solutions optimales Pareto, produit Nash.

Les tableaux suivants montrent tous les résultats de la session :

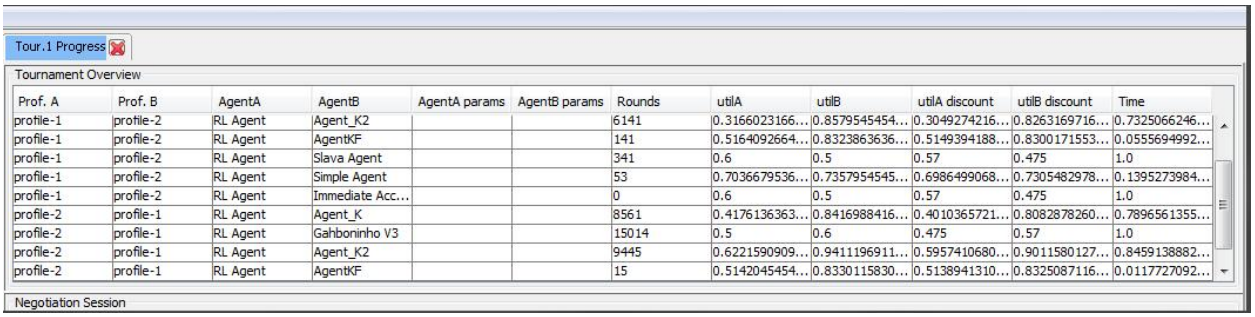

Tour.1 Progress

Tournament Overview

Prof. A	Prof. B	AgentA	AgentB	AgentA params	AgentB params	Rounds	utilA	utilB	utilA discount	utilB discount	Time
profile-1	profile-2	RL Agent	Agent_K			6661	0.3938223938...	0.8877840909...	0.3772104657...	0.8503362320...	0.84020211045
profile-1	profile-2	RL Agent	Gahboninho V3			10500	0.6	0.5	0.57	0.475	1.0
profile-1	profile-2	RL Agent	Agent_K2			6141	0.3166023166...	0.8579545454...	0.3049274216...	0.8263169716...	0.7325066246...
profile-1	profile-2	RL Agent	AgentKF			141	0.5164092664...	0.8323863636...	0.5149394188...	0.8300171553...	0.0555694992...
profile-1	profile-2	RL Agent	Slava Agent			341	0.6	0.5	0.57	0.475	1.0
profile-1	profile-2	RL Agent	Simple Agent			53	0.7036679536...	0.7357954545...	0.6986499068...	0.7305482978...	0.1395273984...
profile-1	profile-2	RL Agent	Immediate Acc...			0	0.6	0.5	0.57	0.475	1.0
profile-2	profile-1	RL Agent	Agent_K			8561	0.4176136363...	0.8416988416...	0.4010365721...	0.8082878260...	0.7896561355...
profile-2	profile-1	RL Agent	Gahboninho V3			15014	0.5	0.6	0.475	0.57	1.0

Negotiation Session

Figure 22. Tableau de résultats.


Tour.1 Progress

Tournament Overview

Prof. A	Prof. B	AgentA	AgentB	AgentA params	AgentB params	Rounds	utilA	utilB	utilA discount	utilB discount	Time
profile-1	profile-2	RL Agent	Agent_K2			6141	0.3166023166...	0.8579545454...	0.3049274216...	0.8263169716...	0.7325066246...
profile-1	profile-2	RL Agent	AgentKF			141	0.5164092664...	0.8323863636...	0.5149394188...	0.8300171553...	0.0555694992...
profile-1	profile-2	RL Agent	Slava Agent			341	0.6	0.5	0.57	0.475	1.0
profile-1	profile-2	RL Agent	Simple Agent			53	0.7036679536...	0.7357954545...	0.6986499068...	0.7305482978...	0.1395273984...
profile-1	profile-2	RL Agent	Immediate Acc...			0	0.6	0.5	0.57	0.475	1.0
profile-2	profile-1	RL Agent	Agent_K			8561	0.4176136363...	0.8416988416...	0.4010365721...	0.8082878260...	0.7896561355...
profile-2	profile-1	RL Agent	Gahboninho V3			15014	0.5	0.6	0.475	0.57	1.0
profile-2	profile-1	RL Agent	Agent_K2			9445	0.6221590909...	0.9411196911...	0.5957410680...	0.9011580127...	0.8459138882...
profile-2	profile-1	RL Agent	AgentKF			15	0.5142045454...	0.8330115830...	0.5138941310...	0.8325087116...	0.0117727092...

Negotiation Session

Figure 23. Suite tableau de résultats.

Nous avons testé notre agent « RL agent » par tous les scénarios non linéaires disponibles dans GENIUS avec tous les agents compatibles avec les utilités non linéaires sous forme d'un tournoi.

Conclusion :

Dans ce chapitre, nous avons testé notre agent «RL agent » dans la plateforme GENIUS avec différents agents qui sont compatibles avec les utilités non linéaire pour les paramètres d'apprentissage dans une négociation multi-issues non linéaire. .

Enfin, ces travaux nous ont aidés à finaliser la conception d'un agent négociateur par SARSA.

conclusion

Conclusion générale :

Le point que nous avons adoptées au cours de ce mémoire pour la conception de notre agent négociateur est celui qui agit de manière intelligente et apprenait les veux de leur adversaire par SARSA au but de maximiser leur utilité toujours et gagner le tour de négociation, nous testons notre agent «RL agent » sur la plateforme GENIUS est nous obtenons des résultats satisfaisantes chez notre agent et l'adversaire.

Références

[1]-Benhmida,F .«Évaluation des Performonces des systémes multi-agents, Thése de doctorat en informatique»,Université BordeauxI ,France ,2013/2014.

[2]- Ferber, J. (1997). Les systèmes multi-agents : un aperçu général. *Technique et Science Informatiques*, pages 979–1012.

[3]-Ferber, J. (1995). *Les Systèmes multi-agents : vers une intelligence collective*. InterEditions.

[4]-. Demazeau, Y. and Costa, A. C. R. (1996). Populations and organizations in open multi-agent systems. In *1st National Symposium on Parallel and Distributed AI*, pages 1–13.

[5]- Wooldridge, M., Jennings, N. R., and Kinny, D. (2000). The gaia methodology for agentoriented analysis and design. *Journal of Autonomous Agents and Multiagent Systems (AAMAS)*, 3 :285–312.

[6]-Russell and Norvig, 2009 Russell, S. J. and Norvig, P. (2009). *Artifificial Intelligence : A Modern Approach (3rd Edition)*, chapter number 2 : Intelligent agents, pages 34–59. Prentice Hall.

[7]- Chin, K. O., Gan, K. S., Alfred, R., Anthony, P., and Lukose, D. (2014). Agent architecture : An overview. *Transactions on Science and Technology*, 1(1) :18–35

[8]- Wood, M. and Deloach, S. A. (2000). An overview of the multiagent systems engineering methodology. In *The First International Workshop on Agent-Oriented software Engineering (AOSE)*, pages 207– 222. Springer.

[9]- Georgeffff, M. P., Pell, B., Pollack, M. E., Tambe, M., and Wooldridge, M. 1999. The beliefdesire-intention model of agency. In *5th International Workshop onIntelligent Agents V, Agent Theories, Architectures, and Languages*, pages 1–10, London, UK, UK. Springer.

[10]- Kinny, D., Georgeffff, M., and Rao, A. (1996). A methodology and modelling technique for systems of bdi agents. In *European Workshop on Modelling Autonomous Agents in a Multi-agent World : Agents Breaking Away : Agents Breaking Away (MAAMAW)*, pages 56–71, Secaucus, NJ, USA. Springer.

[11]- Brazier, F. M. T., Dunin-keplicz, B. M., and Jennings, N. R. (1997). Desire : Modeling multiagent systems in a compositional formal framework. In *International*

Journal of Cooperative Information Systems, volume 6, pages 67–94. World Scientifific Publishing Co.

[12]- Brazier, F. M. T., Jonker, C. M., and Treur, J. (2002). Principles of component-based design of intelligent agents. *Data and Knowledge Engineering*, 41(1) :1–27.

[13]-. Ferber, J. and Gutknecht, O. (1998). A meta-model for the analysis and design of organizations in multi-agent systems. pages 128–135, Paris, France. IEEE Computer Society Press.

[14]- Jamont, J.-P. and Occello, M. (2007). Designing embedded collective systems : The diamond multiagent method. In *19th IEEE International Conference on Tools with Artifificial Intelligence (ICTAI)*, pages 91–94, Washington, DC, USA. IEEE Computer Society Press.

[15]-Kubera, Y., Mathieu, P., and Picault, S. (2010). Everything can be agent ! pages 1547– 1548. International Foundation for Autonomous Agents and Multiagent Systems.

[16]. Demazeau, Y. (1995). From interactions to collective behaviour in agent-based systems. In *1st European Conference on Cognitive Science*, pages 117–132.

[17] J.amont, J.-P. (2005). *DIAMOND : Une approche pour la conception de systèmes multi-agents embarqués*. PhD thesis, Institut National Polytechnique de Grenoble - INPG.

[18]Kubera, Y., Mathieu, P., and Picault, S. (2011). Ioda : An interaction-oriented approach for multi-agent based simulations. *Journal of Autonomous Agents and Multiagent Systems (AAMAS)*, 23(3) :303–343.

[19] LIFL (2011). Ioda ants.http://www.lifl.fr/SMAC/projects/ioda/ioda/models/iodaants/index. html. Accessed : 2022-04-16.

[20]-Jagga, A., Juneja, D., and Singh, A. (2015). Updates in knowledge query manipulation language for complex multiagent systems. *International Journal of Computing Academic Research (IJCAR)*, 4(1) :19 – 26.

[*21*]*Zerrougui ,A.* Elicitation des préférences dans un domaine de négociation multi-issues non linéaire,memoire de master,*Université Larbi Tébessi , Tébessa,2020.*

[22]FIPA (2002). Fipa standard status specififications. http://www.fipa.org/repository/ standardspecs.html. Accessed : 2022-04-16.

[23]Smarsly, K., Law, K. H., and Hartmann, D. (2012). A multi-agent-based collaborative framework for a self-managing structural health monitoring system. *Journal of Computing in Civil Engineering*, 26(1) :76–89.

[24]Poslad, S. (2007). Specifying protocols for multi-agent systems interaction. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 2(4) :51–75.

[25][consortium, 2003] consortium, F. (2003). *FIPA Communicative Act Library, Specifification and FIPA ACL Message Structure Specification*. Technical report.

[26]Bergenti, F. and Ricci, A. (2002). Three approaches to the coordination of multiagent systems. In *2002 ACM Symposium on Applied Computing (SAC)*, pages 367–372, New York, NY, USA. ACM.

[27]Subagdja, B. and Tan, A.-H. (2014). On coordinating pervasive persuasive agents. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1467–1468, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

[28]Coates, G., Whitfifield, R. I., Duffffy, A. H. B., and Hills, B. (2000). Coordination approaches and systems – part ii : An operational perspective. *Research in En*[18] -- «Critères d'évaluation , Tecnologies de l'information et de la communication et appropriation des savoirs ; Agent intelligents cours Web interactif »[Enligne].Available :http ://turing.cs.pub.ro/auf2/html/chapters/chapter5/chapter_5_2_2. html. [Accès le 01-06-2022]. *gineering Design*, 12(2) :73 – 89.

[29]A. Florea, «agents intelligent cour web interactif, chapitre 2: architecture des agents et language,» Politechnia University of Bucharest, 2002.29

[30]N.Bakhta, «Modèle multi agents pour la conception de système d'aide à la décision collective, Thése de doctorat en informatique,» université d'oran , oran, 2013/2014.

[31]T. MARIR, «les système multi agents,» oum lbouaghi, unversité larbi Ben Mhidi, 2016/2017.

[32]S, Mourad et C, Brahim. « *Stratégies de négociation entre agents dans le domaine du transport* ». Département d'Informatique, Faculté des Sciences et de Génie, Université Laval, Ste-Foy, PQ, Canada, G1K 7P4.

[33]J. Jamont, «Démarche modèle et outils pour l'ingenierie des collectifs cyber

physiques, Memoire,» Université Grenoble alpes, 2016.

[34]B. samir, «une approche multi agent d'identification précoce des interactions entre les aspects,» skikda, juin 2009.

[3]5M. H. Verrons, «GENAC: un modèle générale de négociation de contrats entre agent.,» 12 jan 2005.

[36«]Critères d'évaluation , Tecnologies de l'information et de la communication et appropriation des savoirs ; Agent intelligents cours Web interactif » [Enligne].Available :http ://turing.cs.pub.ro/auf2/html/chapters/chapter5/chapter_5_2_2. html. [Accès le 28 02 2020].

[37]M. Morgue, «Contributions à la négociation multi agents,» université de Lille, 2020.

[38]A. Mansour « Apprentissage profond(Deep Learning) pour la classification des lames anapath numérisées,Mémoire pour l'obtention de master 2 » tebessa , Univérsité Echikh larbi tebessi ,2019.

[39]Matthieu Geist. Optimisation des chaînes de production dans l'industrie sidérurgique : une approche statistique de l'apprentissage par renforcement. Mathématiques générales [math.GM]. Université Paul Verlaine - Metz, 2009. Français. ffNNT : 2009METZ023Sff. fftel-01752647v1

[40]A. Mansour « Apprentissage profond(Deep Learning) pour la classification des lames anapath numérisées,Mémoire pour l'obtention de master 2 » tebessa , Univérsité Echikh larbi tebessi ,2019.

[41]S. K. e. a. RAZ LIN, «G E N I U S: AN INTEGRATED ENVIRONMENT FOR SUPPORTING THE DESIGN OF GENERIC AUTOMATED NEGOTIATORS,» *computinal intelligent,* p. 23, 2012.

[42]«tUDelfet, ANAC International negotiation competition » [En ligne]. Available: http://ii.tudelft.nl/genius/?q=article/releases. [Accès le 20 05 2022].

[43]« Introduction à Eclipse, cour licence informatique » Université de Lille 1 -2013-2014.

[44]o. s. e. fréderik, «chapitre3 :apprentissage par renforcement».