République Algérienne Démocratique et Populaire

Ministère de l’Enseignement Supérieur et de la Recherche Scientifique

**Université des Sciences et de la Technologie Houari Boumediene**

Département Informatique

# Module : Technologie des agents

**Rapport du projet Technologie des agents**

**Réalisation par :**

**-MESSARAT Mehdi 201400007400**

**-BEZGALI Meriem 201300005476**

**Groupe : 01**

1. **Introduction :**

L'[intelligence](https://www.futura-sciences.com/sante/definitions/corps-humain-intelligence-13498/) artificielle (IA, ou AI en anglais pour Artificial Intelligence) consiste à mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme d'intelligence réelle. L'IA se retrouve implémentée dans un nombre grandissant de domaines d'[application](https://www.futura-sciences.com/sciences/definitions/mathematiques-application-13200/) comme les systèmes experts et système multi-agents.

L’une des branches que nous avons étudiées dans le module « technologie des agents » est la branche des systèmes multi-agents, ils présentent un grand intérêt pour les applications où les connaissances et le contrôle sont naturellement distribués.

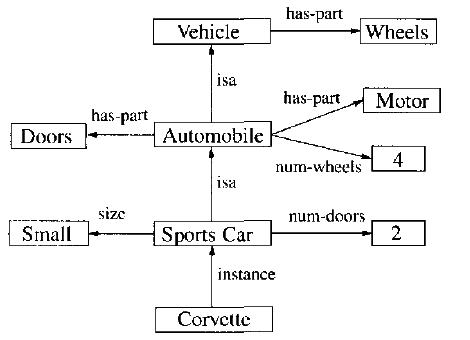
Le but de ce projet est d’appliquer les notions théoriques apprises concernant les systèmes experts et systèmes multi-agents, en concevant au préalable un système experts, puis un système multi-agents comportant des agents intelligents interagissant dans un environnement de vente.

1. **Partie I : Systèmes expert :**

Dans cette partie, nous avons étudié le system expert du livre « Constructing Intelligent Agents Using Java » fournit lors du Tp, ce système expert a une base de règles qui permet d’identifié le type d’un véhicules et qui est définie comme suit :

|  |  |
| --- | --- |
| Bicycle : If vehiculeType=cycle  and num\_wheels=2  and motor=no   then vehicule=Bicycle  Tricycle : If vehiculeType=cycle  and num\_wheels=3  and motor=no   then vehicule =Tricycle  Motorcycle : If vehiculeType=cycle  and num\_wheels=3  and motor=yes  then vehicule = Motorcycle  SportsCar : If vehiculeType=automobile  size=small  num\_doors=2  then vehicule = SportsCar  Sadan : If vehiculeType=automobile  size=medium  num\_doors=4  then vehicule = Sadan | MiniVan : If vehiculeType=automobile  size=medium  num\_doors=3  then vehicule = MiniVan  SUV : If vehiculeType=automobile  size=large  num\_doors=4  then vehicule = SUV  Cycle : If num\_wheels<4   then vehicule = Cycle  Automobile : If num\_wheels=4   and motor =yes  then vehicule = automobile |

D’un point de vue sémantique le schéma ci-dessous illustre le lien entre les types de véhicule.

  
**figure1.**un exemple sémantique sous forme de réseau

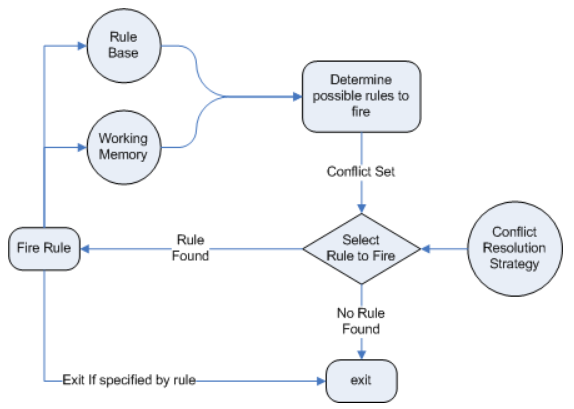
1. **Définition d’un système expert :**

Un système expert est un outil capable de reproduire les mécanismes cognitifs d'un expert, dans un domaine particulier. Il s'agit de l'une des voies tentant d'aboutir à l'intelligence artificielle, et plus précisément, un système expert est un logiciel capable de répondre à des questions, en effectuant un raisonnement à partir de faits et de règles connues. Il peut servir notamment comme outil d'aide à la décision.

Un système expert se compose de 3 parties :

1. Une base de faits: est l'une des entrées d'un moteur d'inférence. C'est un ensemble de connaissances appelées "faits" et considérés comme vrais. À partir de ces faits, le moteur d'inférence va leur appliquer les règles issues de sa base de règles pour en déduire d'autres faits et ainsi résoudre un problème.
2. Une base de règles: une base de règles regroupe des règles spécifiques à un domaine spécialisé donné, sous une forme exploitable par un ordinateur.
3. Un moteur d’inférence : Le moteur d'inférence est un mécanisme qui permet d'inférer des connaissances nouvelles à partir de la base des faits et la base des règles. Il est basé sur des règles d'inférence qui régissent son fonctionnement. Il a pour fonction de répondre à une requête de la part d'un utilisateur ou d'un serveur. Il peut fonctionner en chaînage avant, chaînage arrière.

Et à coté du moteur d’inférence il existe un mécanisme permettant de résoudre les conflits s’il existe plusieurs règles applicable, le schéma suivant illustre de manière globale le fonctionnement d’un système expert et est valable pour les systèmes experts que nous allons aborder un peu plus tard.

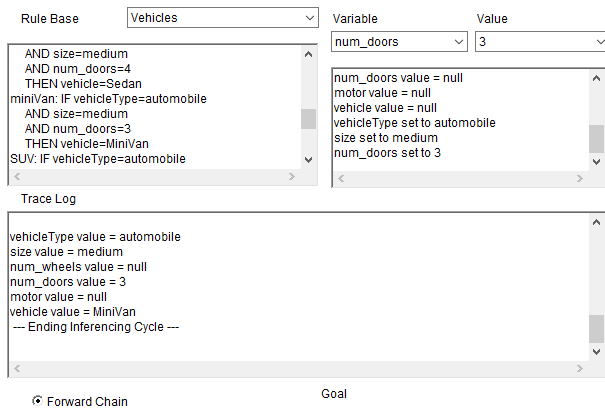


**Figure2.**schéma du fonctionnement d’un système expert

1. **Chainage avant :**

Les moteurs à chaînage avant démarrent des faits et règles, et tentent de s'approcher des faits recherchés par le problème.

Ci-dessous un exemple de chainage avant en utilisant la base de règles « vehicule type » implémentée dans la question 1 du projet, nous allons inférer le type « Mini-Van » à partir des faits = { VehiculeType =automobile**,** num\_doors=3, size =meduim }

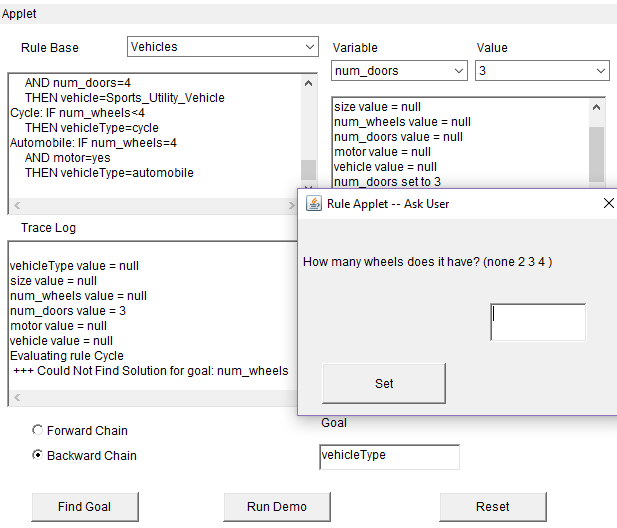


**Figure3.**exécution de l’inférence avec chainage avant

1. **Chainage arrière :**

Les moteurs à chaînage arrière partent des faits recherchés par le problème, et tentent par l'intermédiaire des règles, de remonter à des faits connus.

Ci-dessous un exemple de chainage arrière en utilisant la base de règles « vehicule type » implémentée dans la question 1 du projet, nous allons inférer le type « Mini-Van » à partir des faits : {num\_doors=3} et en précisant le but à « vehicleType ».



**Figure4.**exécution de l’inférence avec chainage arrière

Remarquer que par chainage arrière si on manque de faits pour obtenir le but, un mécanisme est implémenté qui permet de demander à l’utilisateur d’introduire certain fait afin d’aidé à l’inférence du but voulu.

**II.2. Implémentation des règles en se basant sur le système expert existant :**

Nous avons choisi d’adapter le système précédent pour traiter le domaine de vente de vêtements, afin de mieux comprendre le fonctionnement des systèmes.

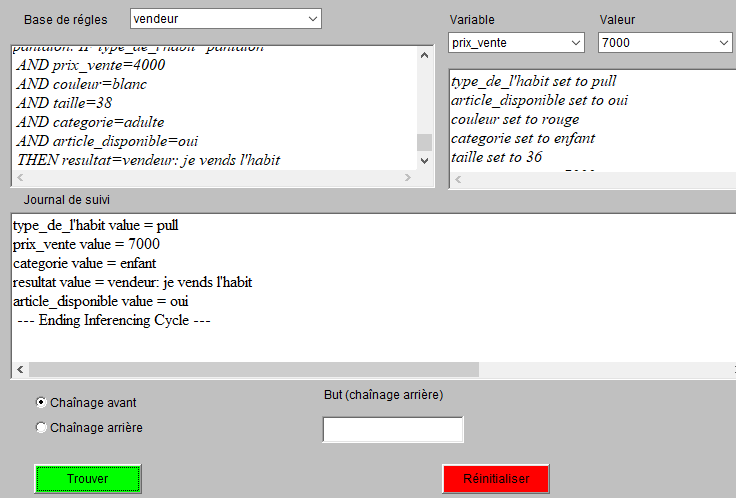
Un magasin propose à ses clients des vêtements caractérisés par le prix, la taille, types, couleur et disponibilité de l’article ou non.

* Un acheteur se base sur ses préférences sur chacun des caractéristiques afin de prendre une décision d’achat.
* Le vendeur pourra voir les articles qui sont en disponibilité dans le magasin, et ainsi faire la décision de vendre l’article selon sa disponibilité chez lui.

Il s’agit de construire un système expert qui répond aux questions posées par l’acheteur et le vendeur en se basant sur leurs préférences afin de les aider à prendre leurs décisions, On notera qu’ici les préférences représentent les faits et que la base de règles est citée en détail en annexe.

1. **Exemple chainage avant :**

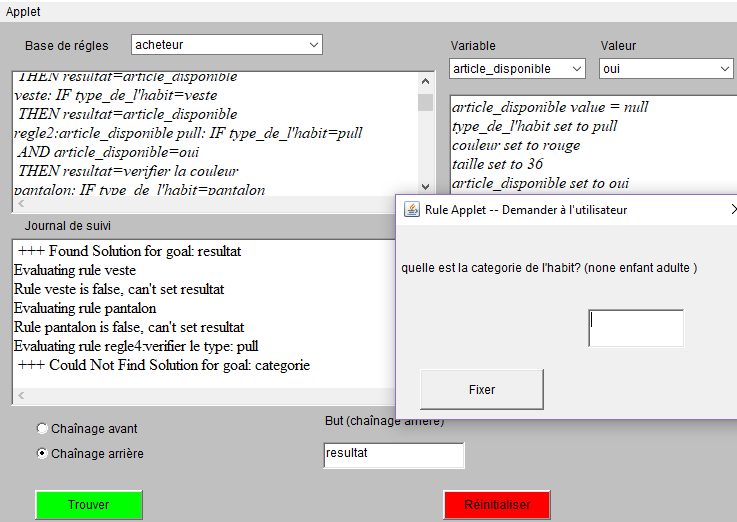
Nous allons inférer la décision du vendeur en utilisant la base de faits suivante:  
Base de faits = {type\_de\_l'habit = pull, article\_disponible=oui, couleur=rouge, taille= 36, prix\_vente=7000, categorie=enfant}

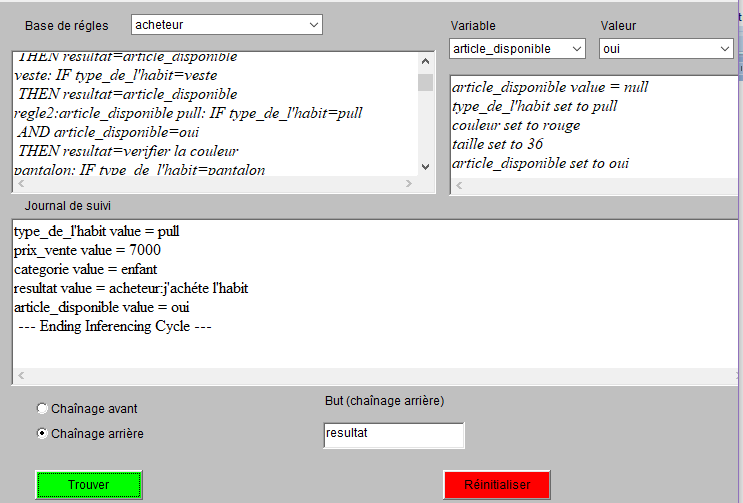
**Figure5.**exécution de l’inférence avec chainage avant

Remarque: on a aussi implémenté une règle qui fait appelle à une autre règle.

1. **Exemple chainage arrière :**

Nous allons inférer la décision de l’acheteur en utilisant la base de faits suivante:  
Base de faits = {type\_de\_l'habit = pull, article\_disponible=oui, couleur=rouge, taille= 36} et comme but « résultat » qui est la décision de l’acheteur.

**Figure6.**exécution de l’inférence avec chainage arrière

 **Figure7.**suite de l’exécution de l’inférence avec chainage arrière

On remarque qu’il manque des faits au SE pour faire l’inférence, donc on demande à l’utilisateur de les introduire si possible afin de continuer l’inférence et déduire le but voulu.

**II.3. Conclusion :**

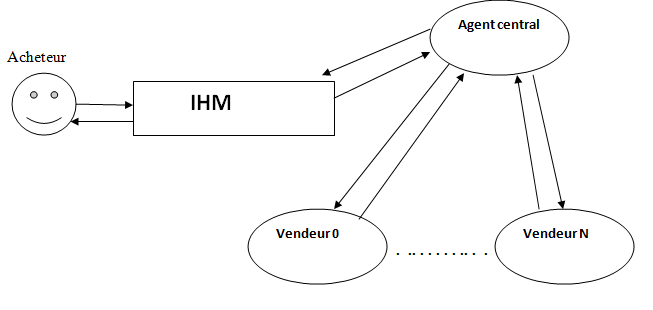
Cette partie du projet nous a aidé à mieux comprendre les systèmes experts et leurs fonctionnements grâce à la première question qui nous a permit de nous familiarisé avec un SE et par la suite de mettre en pratique ce que nous avons appris en faisant la deuxième question.

1. **Partie II : Implémentation d’un environnement de vente à base d’agents intelligents :**

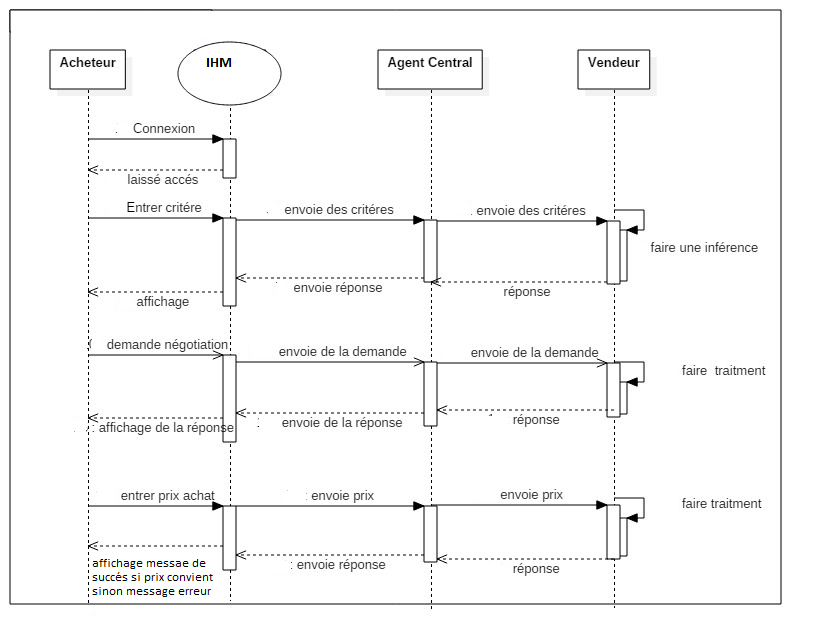
Dans cette partie nous avons implémenté un environnement de vente décentralisé à base d’agent intelligent et qui comporte les acteurs suivant:

|  |  |
| --- | --- |
| Acteur | Description et rôle |
| -un agent central | Intermédiaire entre l’acheteur et le vendeur qui effectue les taches suivantes :   * Transmettre les messages et les critères de l’acheteur aux vendeurs. * Transmettre les réponses des vendeurs à l’acheteur. * Créer les agents vendeurs si article disponible dans le magasin. * Transmettre la requête de négociation de l’article au vendeur et envoie la réponse à l’acheteur. |
| -des agents vendeurs | vendent des vêtements d’une catégorie bien précise selon des critères que l’acheteur doit spécifier, et qui effectue les taches suivantes :   * Attendre les requêtes de l’acheteur de la part de l’agent central. * Répondre aux requêtes de l’acheteur en envoyant sa réponse à l’agent central. |
| -un acheteur | l’utilisateur qui utilise notre environnement de vente et qui effectue les taches suivantes :   * Utilise l’environnement de vente pour effectuer son achat. * Peut envoyer une demande de négociation du prix. * Communique avec l’agent central. * Entre les attributs recherchées dans l’environnement pour voir la disponibilité d’un article et l’acheté éventuellement. |
| -gestionnaire | responsable d’ajout des catégories et qui effectue les taches suivantes   * Ajouter une nouvelle catégorie. * Mettre à jour une catégorie déjà existante. * Supprimer une catégorie. |

De cette description et des rôles des acteurs on peut déduire le schéma du système multi-agents qui se présente de la sorte :

 **Figure8.**Architecture de l’SMA

Ainsi que le diagramme de séquence qui montre les étapes du déroulement d’une vente et les messages envoyés par chaque acteur :



**Figure9.**Diagramme de séquence d’un scénario de vente

**III.1. Implémentation et environnement de travail :**

1. **Environnement de travail :**

Afin d’implémenter ce SMA nous avons utilisé le langage JADE qui est une plateforme de [programmation](https://fr.wikipedia.org/wiki/Programmation_informatique) multi-agent implémentée en [Java](https://fr.wikipedia.org/wiki/Java_(langage)), les agents qui tournent sous JADE communiquent via des messages avec le langage Agent Communication Language (ACL)

Quant à l’environnement de développement nous avons utilisé **Eclipse** (version Luna) open source, qui est un environnement pragmatique, facile a utilisé et très riche en extension et plugin qui sont disponible en téléchargement gratuit.

1. **Implémentation :**

Il nous a été demandé d’implémenté un SMA dont les agents vendeurs possèdent un système expert à base de règles, nous avons choisit de faire les règles sur 4 attributs suivants:

-la couleur.   
- la taille.   
- la marque.   
-la catégorie.

Nous avons fait la combinaison de ces 4 attributs pour obtenir un système expert basé sur 16 règles.

Exemple d’une règle:

« Si couleur\_voulue = couleur existante et taille\_voulu = taille existe et marque\_voulue = marque\_existante alors transmettre message « article existe » à l’agent central »

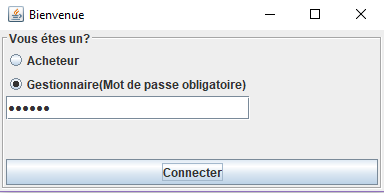
On a aussi ajouté la possibilité d’ajout de nouvelle catégorie par l’acteur « gestionnaire » grâce à une interface conçu spécialement pour lui que nous allons voir par la suite.

1. **Interface et exécution :**

Nous allons maintenait procédé à la présentation des différentes interfaces de l’application.

1. **Interface de connexion :**

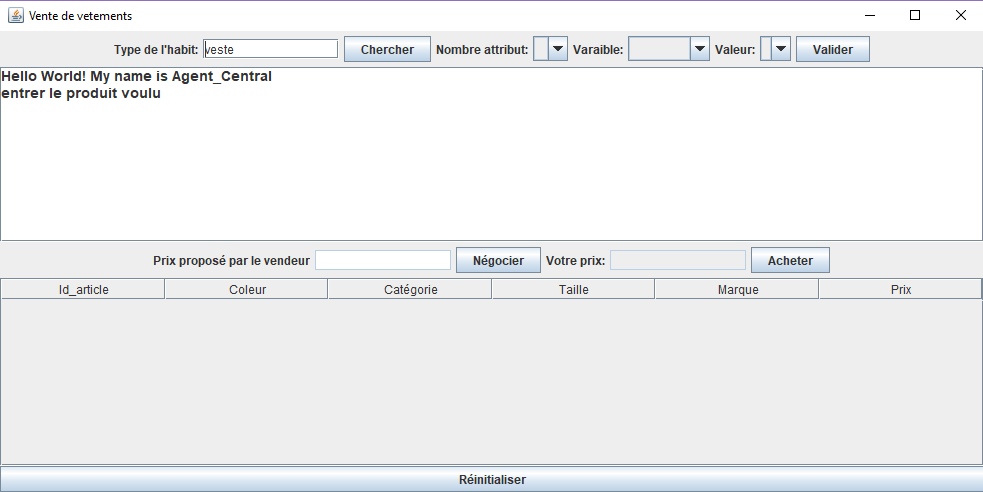
Permet à l’acheteur de se connecter pour effectuer son achat et au gestionnaire de se connecter pour mettre à jour les différentes catégories, le mot de passe est obligatoire pour l’identification du gestionnaire.



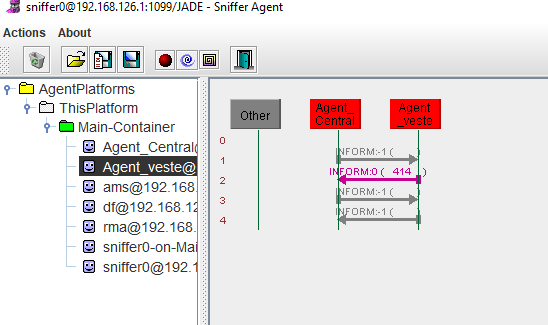
**Figure10.**interface de connexion

1. **Interface de vente :**

Après sa connexion, l’acheteur est dirigé vers l’interface « vente de vêtements». Sur cette fenêtre, un acheteur peut effectuer un achat. Il choisit un type d’habit, le nombre d’attribut qu’il veut insérer pour effectuer sa recherche, puis il spécifie les variables et leurs valeurs, enfin il valide. Par la suite l’agent central prend les critères entrés et les transmet à l’agent vendeur créé préalablement.

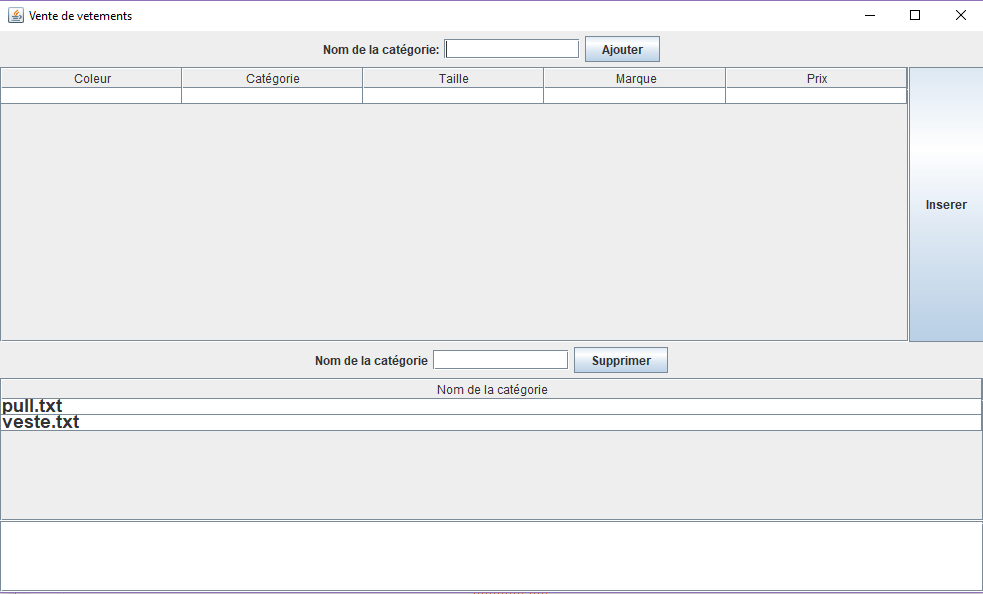
 **Figure11.**interface de vente

La figure suivante montre les messages échangés entre un agent vendeur et l’agent central suite à la validation des attributs par l’acheteur et demande de négociation du prix de vente.

 **Figure12.**lancement du sniffing

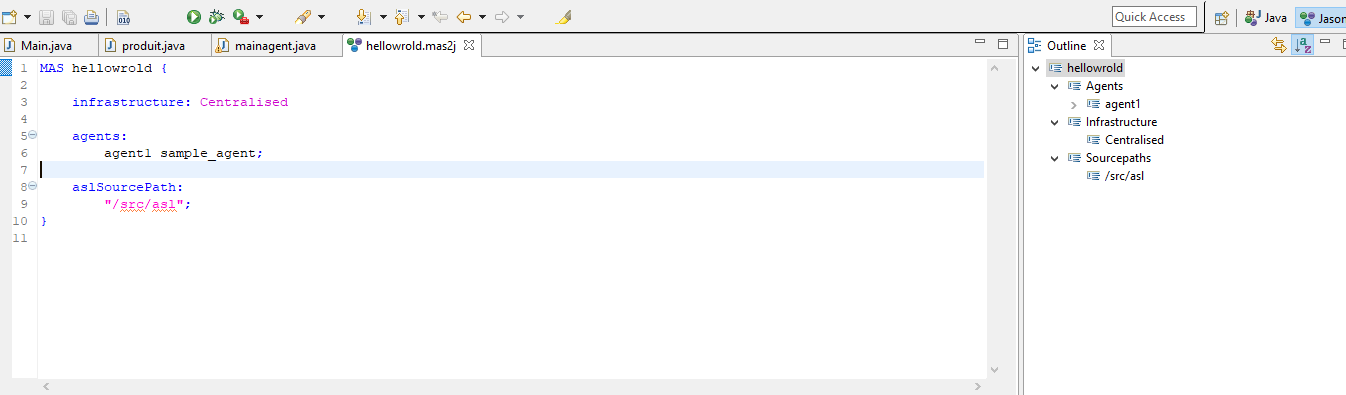
1. **Interface d’ajout de catégorie et de leurs mises à jour :**

Suite à la connexion du gestionnaire ce dernier est dirigé sur l’interface de mise à jour des catégories ou il peut ajouter une nouvelle catégorie, en supprimer une et la mettre à jour, il les remplit éventuellement en insérant les lignes des produits composant une catégorie.

 **Figure13.**interface mise à jour des catégories

1. **Partie III : Implémentation d’un environnement de vente à base d’agents intelligents avec la plateforme Jason:**

Nous avons téléchargé et installé le plugin Jason sous « Ecplise » et nous avons testé un programme simple juste pour pouvoir faire la comparaison entre les deux plateformes Jade et Jason.

**Figure14.**Plateforme Jason

1. **Comparaison :**

L'une des approches les plus connues du développement des agents cognitifs est l'architecture BDI (Beliefs-Desires-Intentions) qui convient parfaitement à la plateforme Jason, un agent implémenté sous Jason a la capacité d'effectuer des cycles de raisonnement, c'est-à-dire son architecture BDI effectue un cycle de sens-plan-acte, qui lui permet d’évaluer les plans qui sont déclenchés par l’arrivé d’un événement se qui fait la différence avec JADE, qui à la place, fournit l'abstraction de l'agent comme un ensemble de comportements avec des capacités de communication suite à l’arrivé d’un événement précis. Alors que sous Jason il n’est pas nécessaire pour les agents de prévoir un traitement spécifique des différents événements perçu, dans faits, ils peuvent être modélisés comme des événements Jason réguliers, alors que sur Jade le programmeur de l’agent va implémenter la logique de gestion des événements, en ajoutant des comportements appropriés au référentiel de comportement de l'agent, quand planifié, le comportement sera exécuté et l'événement sera géré et donc Jade est beaucoup plus orienté agent réactif et Jason est conçu pour créer des agents cognitifs.

1. **Conclusion :**

Ce projet nous a permis d’apprendre énormément sur les systèmes experts et nous a été d’un grand apport en matière de conception de systèmes multi-agents.

Nous avons vu le rôle des systèmes experts dans la modélisation des situations du monde réel de manière concise et d’un autre côté nous avons pris conscience de l’importance des systèmes multi-agents.   
Nous nous sommes aussi familiarisé avec les deux plateformes Jason et Jade et nous avons appris à les utilisés et a comprendre la différence entre eux et le type d’agent qu’ils permettent d’implémenté, Jason agent cognitif et Jade agent réactif.

**Annexe**

**Vendeur Rule Base :**

regle1: verifie le type\_de\_l'habit: pull: IF type\_de\_l'habit=pull THEN resultat = article disponible

pantalon: IF type\_de\_l'habit=pantalon THEN resultat=article\_disponible

veste: IF type\_de\_l'habit=veste THEN resultat=article\_disponible

regle2:article\_disponible pull: IF type\_de\_l'habit=pull AND article\_disponible=ouiTHEN resultat=verifier la couleur

pantalon: IF type\_de\_l'habit=pantalon AND article\_disponible=oui THEN resultat=verifier la couleur

veste: IF type\_de\_l'habit=veste AND article\_disponible=oui THEN resultat=verifier la couleur

regle3:verifier la couleur pull: IF type\_de\_l'habit=pull AND couleur=rouge AND article\_disponible=oui THEN resultat=verifier la categorie

veste: IF type\_de\_l'habit=veste AND couleur=vert AND article\_disponible=oui  THEN resultat=verifier la categorie

pantalon: IF type\_de\_l'habit=pantalon AND couleur=blanc AND article\_disponible=oui  THEN resultat=verifier la categorie

regle4:verifier le type: pull: IF type\_de\_l'habit=pull AND couleur=rouge AND article\_disponible=oui AND categorie=enfant THEN resultat=verifier la taille

pantalon: IF type\_de\_l'habit=pantalon AND article\_disponible=oui AND couleur=blanc AND categorie=adulte THEN resultat=verifier la taille

veste: IF type\_de\_l'habit=veste AND article\_disponible=oui AND couleur=vert AND categorie=adulte THEN resultat=verifier la taille

regle5:verifier la taille pull: IF type\_de\_l'habit=pull AND couleur=rouge AND categorie=enfant AND article\_disponible=oui AND taille=36 THEN resultat=prix\_vente   
pantalon: IF type\_de\_l'habit=pantalon AND article\_disponible=oui   
AND couleur=blanc AND taille=38 AND categorie=adulte THEN resultat=prix\_vente

veste: IF type\_de\_l'habit=veste AND article\_disponible=oui   
 AND couleur=vert AND taille=38 AND categorie=adulte   
THEN resultat=prix\_vente

régle6:prix\_vente pull: IF type\_de\_l'habit=pull AND prix\_vente=7000 AND couleur=rouge AND taille=36 AND categorie=enfant AND article\_disponible=oui THEN resultat=vendeur: je vends l'habit   
veste: IF type\_de\_l'habit=veste AND prix\_vente=8000 AND couleur=vert AND taille=38 AND categorie=adulte AND article\_disponible=oui THEN resultat=vendeur: je vends l'habit

pantalon: IF type\_de\_l'habit=pantalon AND prix\_vente=4000 AND couleur=blanc AND taille=38 AND categorie=adulte AND article\_disponible=oui THEN resultat=vendeur: je vends l'habit