



République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
**Université des Sciences et de la Technologie Houari Boumediene**



Faculté d'Informatique

**Projet Technologie des agents**

Filière: Informatique

Spécialité: SII

Conception et mise en place d'un système expert et multi agents pour une application  
e-commerce

**Présenté par :**

**DAHMANI**

Meriem

**BOULMAALI**

Linda Imene

# **Sommaire**

## **Introduction générale**

### **Partie 1: Construction et mise en place d'un système expert**

1. Qu'est Ce qu'un système système expert ?
2. Approche choisie.
3. Les tests effectués.

### **Partie 2 Construction et mise en place d'un système multi agents**

1. Qu'est Ce qu'un système multi agents ?
2. Approche choisie pour la conception
3. Outils choisis pour l'implémentation
4. Test

## **Conclusion Générale**

## **Références bibliographiques**

## **Introduction**

Dans ce projet nous allons mettre en œuvre un système expert et un système multi-agent pour une application desktop e-commerce. Le but de l'application est de donner la meilleure expérience possible au client et de lui fournir le produit voulu dans un temps raisonnable et pour cela, nous allons faire les tests sur deux systèmes différents.

Dans la première partie, nous allons implémenter un système expert afin de simuler une vente de produits informatiques, et une autre simulation de l'exemple de voiture demandé dans l'énoncé.

Dans la seconde partie nous allons travailler sur un système multi agents qui interagit avec le client et lui fournit des produits à partir de différents magasins.

Nous allons commencer notre simulation par la conception de nos systèmes, l'implémentation des raisonneurs et enfin les tests.

# Partie 1 : Construction et mise en place d'un système expert

## Introduction

Dans cette partie du projet, nous allons mettre en œuvre le système expert.

## Qu'est Ce qu'un système système expert ?

Le système expert est un outil conçu pour modéliser le raisonnement d'un spécialiste humain, en reproduisant ses mécanismes cognitifs. Le progiciel effectue ainsi un raisonnement à partir de faits connus, et de règles implémentées au départ. Ainsi, il peut réaliser des déductions ou conclusions et des explications sur la manière dont les résultats ont été obtenus.[1]

## Approche choisie

Chaque système expert de notre application contient une base de règles qui fournit le produit voulu selon les choix effectués par l'utilisateur et qui sont représentés par une base de faits (bdf). A chaque étape, on essaie de déduire plus de caractéristiques sur le produit et pour cela on applique des règles en allant des faits prédéfinis jusqu'à atteindre le produit approprié s'il existe dans notre base.

Dans le cas où on aura plusieurs règles possibles à appliquer, on définit une approche qui permet de choisir une règle parmi plusieurs présenté dans conflict set, Cette approche permet de sélectionner la règle qui contient le plus de clauses dans la partie des prérequis.

On présente dans ce qui suit la base de connaissance de chaque système expert:

- **Le système des véhicules:**

<p><b>Bicycle:</b> If vehicleType=Cycle and num_wheels=2 and motor=no Then vehicle=Bicycle</p> <p><b>Tricycle:</b> If vehicleType=Cycle and and motor=no Then vehicle=Tricycle</p> <p><b>Motorcycle:</b> If vehicleTvpe=Cycle and and motor=yes Then vehicle=Motorcycle</p> <p><b>SportCar:</b> If vehicleType=automobile and size=small and Then vehicle-SportCar</p> <p><b>Sedan:</b> If vehicleType-automobile and size-medium and num_doors=4 Then vehicle-Sedan</p>
--

**MiniVar:** If vehicleType=automobile and size=medium and Then  
 vehicle=MiniVar  
**SUV:** If vehicleType=automobile and size=large and Then  
 vehicle=SportUtilityVehicle  
**Cycle:** If num\_wheels < 4 Then vehicleType=Cycle  
**automobile:** If num\_wheels = 4 and motor=yes Then vehicleType=automobile

- **Le système de la vente des pc :**

**règle 1:** utilisation = "gaming" → performance\_minimale = "très performant"  
**règle 2:** utilisation = "professionnel" → performance\_minimale = "Très performant"  
**règle 3:** utilisation = "étudiant" → performance\_minimale = "performant"  
**règle 4:** utilisation = "personnel" → performance\_minimale = "basique"  
**règle 5:** utilisation = "bureautique" → performance\_minimale = "basique"  
**règle 6:** performance\_minimale = "très performant" → disque dur = ssd et ram = "16 GO" et puissance\_processeur = très puissant  
**règle 7:** performance\_minimale = "performant" → disque dur = ssd et ram = "8 GO" et puissance\_processeur = puissance moyenne  
**règle 8:** performance\_minimale = "performant" → disque dur = ssd et ram = "16 GO" et puissance\_processeur = puissance moyenne  
**règle 9:** performance\_minimale = "basique" → disque dur = hdd et ram = "< 8 GO" et puissance\_processeur = "faible"  
**règle 10:** if puissance\_processeur = "très puissant" → processeur = "i7" et carte\_graphique\_minimale = "performante"  
**règle 11:** if puissance\_processeur = "très puissant" et performance\_minimale = "performant" → processeur = "i5" et carte\_graphique\_minimale = "performante"  
**règle 12:** if puissance\_processeur = "puissance moyenne" → processeur = "i5"  
**règle 13:** if puissance\_processeur = "puissance moyenne" → processeur = "i7"  
**règle 14:** if puissance\_processeur = "faible" → processeur = "i3"  
**règle 15:** performance\_minimale = "très performant" → carte\_graphique = "NVIDIA GeForce RTX 3060"  
**règle 16:** performance\_minimale = "performant" → carte\_graphique = "Intel UHD 620"  
**règle 17:** performance\_minimale = "faible" → carte\_graphique = "Intel HD 520"  
**règle 18:** performance\_minimale = "faible" → carte\_graphique = "Intel UHD Graphics"  
**règle 19:** carte\_graphic = "NVIDIA GeForce RTX 3060" and processor = "i7" and taille\_ecran = "15.6" et prix > 150000 (cher) → list\_real\_names = "Asus Tuf Dash F15 DASH-TUF516PR-HN075T PC"

**règle 20:** carte\_graphic="NVIDIA GeForce RTX 3060" and processor= "i5  
"and taille\_ecran=" 15.6" et prix<150000DA (moyen)→ list\_real\_names= HP  
OMEN 15-en1003sf PC Portable GAMING "

**règle 21:** carte\_graphic=" Intel UHD Graphics" and processor= " i5" and  
taille\_ecran="14 " et prix<100 000DA (moyen) → list\_real\_names="Microsoft  
Surface Pro 7 "

**règle 22:** carte\_graphic="Intel UHD 620 " and processor= " i7" and  
taille\_ecran="15,6 " et prix100 000DA<prix<150 000DA(moyen)→  
list\_real\_names=" DELL LATITUDE 5591"

**règle 23:** carte\_graphic=" Intel UHD Graphics" and processor= " i5" and  
taille\_ecran="14 "et prix moyen → list\_real\_names="DELL"

**règle 24:** carte\_graphic=" Intel UHD 620" and processor= " i5"and  
taille\_ecran="20.7 " et prix>150 000(cher)→ list\_real\_names=" Pc Bureau  
HP"

**règle 25:** carte\_graphic=" Intel UHD Graphics" and processor= " i7" and  
taille\_ecran="14" → et 130 000DA<prix<150 000DA (cher)  
ist\_real\_names="Lenovo ThinkPad t490s "

**règle 26:** carte\_graphic="NVIDIA MX 130 2GB " and processor= i5and et  
prix<100 000DA (pas cher) taille\_ecran=" 14" et prix moyen→  
list\_real\_names=" LENOVO THINKPAD X380"

**règle 27:** carte\_graphic=" Intel HD 520" and processor= "i5 " and  
taille\_ecran="14 " → et prix<100 000DA et prix<100 000DA pas cher  
list\_real\_names="HP "

**règle 28:** carte\_graphic="Intel UHD Graphics " and processor= "i3 " and  
taille\_ecran="15.6 " et prix<100 000DA (pas cher)→list\_real\_names=" Acer  
Aspire 3 A315-58-39MW"

### Les tests effectués :

- **Le système des véhicules:**

L'utilisateur commence par remplir le formulaire qui contient les champs nécessaires à déterminer le produit but, le formulaire contient les champs suivants :motor, num\_wheels, num\_doors et size.

Après la validation du choix, le moteur d'inférence cherche les règles possibles à appliquer selon la base des faits représentant les choix de l'utilisateur comme expliqué dans la partie d'avant, et il essaye d'arriver à la variable "véhicule" qui représente notre but.

SE1

SE2

motor

yes

num\_wheels

4

num\_doors

3

size

medium

Register

Le Système Expert Des Véhicules

le conflit set :

vehicleType=automobile et size=medium et num\_doors=3 --> vehicle=MiniVar

la règle déclenchée :

vehicleType=automobile et size=medium et num\_doors=3 --> vehicle=MiniVar

résultat :

motor=yes

size=medium

num\_wheels=4

num\_doors=3

vehicleType=automobile

vehicle=MiniVar

goal achieved !

Next

Figure : test effectué sur le système expert des véhicules

- Le système de la vente des PC:

Dans ce système on effectue le même processus et on utilise le même moteur d'inférence, l'utilisateur commence par remplir le formulaire qui contient les champs suivants: use, prix et taille écran.

SE1

SE2

use

gaming

prix

cher

taille\_ecran

15.6

Register

le conflit set :

Performance\_minimale=très performant --> disque\_dur=ssd et ram=16GO et puissance\_processeur=très performant --> carte\_graphique=NVIDIA GeForce RTX 3060

la règle déclenchée :

Performance\_minimale=très performant --> disque\_dur=ssd et ram=16GO et puissance\_processeur=très performant --> carte\_graphique=NVIDIA GeForce RTX 3060

résultat :

utilisation=gaming

prix=cher

taille\_ecran=15.6

Performance\_minimale=très performant

disque\_dur=ssd

ram=16GO

puissance\_processeur=très puissant

Next

Figure : test effectué sur le système expert de la vente des pc

On peut voir dans ce cas que le conflict set contient plus d'une seule règle alors on doit utiliser une méthode qui permet de sélectionner la règle avec le plus de clauses dans la partie des prérequis et on continue ce processus jusqu'à atteindre la variable but "list real name" qui représente la marque et les caractéristiques du pc trouvé selon les choix sélectionnés.

Implémentation d'un système expert

SE1 SE2

use gaming

prix cher

taille\_ecran 15.6

Register

**le conflict set :**

carte\_graphique=NVIDIA GeForce RTX 3060 et processeur=i7 et taille\_ecran=15.6 et prix=cher --> list

**la règle déclenchée :**

carte\_graphique=NVIDIA GeForce RTX 3060 et processeur=i7 et taille\_ecran=15.6 et prix=cher -->

**résultat :**

puissance\_processeur=très puissant  
 processeur=i7  
 carte\_graphique\_minimale<performante  
 processeur=i5  
 carte\_graphique\_minimale<performante  
 carte\_graphique=NVIDIA GeForce RTX 3060  
 list\_real\_names=Asus Tuf Dash F15 DASH-TUF516PR-HN075T PC  
 goal achieved !

Next

**Figure :** le résultat du test effectué précédemment

Au cas où les options choisies ne correspondent à aucun élément dans notre base de données, on affiche un message de fin de recherche, cependant on peut toujours alimenter notre base en ajoutant d'autres pc.

Implémentation d'un système expert

SE1 SE2

use étudiant

prix pas cher

taille\_ecran 20.7

Register

**le conflict set :**

puissance\_processeur=puissance moyenne --> processeur=i7

**la règle déclenchée :**

puissance\_processeur=puissance moyenne --> processeur=i7

**résultat :**

goal not achieved, we don't have the product !

Next

**Figure :** le résultat du test effectué à la recherche d'un produit inexistant



## Partie 2 : Construction et mise en place d'un système multi agents

### Introduction

Dans cette partie du projet, nous allons mettre en œuvre LE système multi agent.

### Qu'est Ce qu'un système multi agents?

Un système multi agent est un système composé d'un ensemble d'agents (un processus, un robot, un être humain, une fourmi, etc.), actifs dans un certain environnement et interagissant selon certaines règles. Un agent est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome, ce qui exclut un pilotage centralisé du système global.[2]

### Approche choisie

La base de connaissance de chaque agent a été faite à l'aide d'un fichier json, structuré comme suite:

```
{
  "magasin":"nom_magasin",
  "ventes":[
    {
      "categorie":"categorie_name",
      "categorie_name":[
        {
          "id":1,
          "marque": "samsung",
          "taille":8,
          "quantity":5,
          "prix":7000

          // d'autres caractéristiques_pour_chaque_catégorie
        }
      ]
    }
  ]
}
```

Les agents ont plusieurs articles en commun.

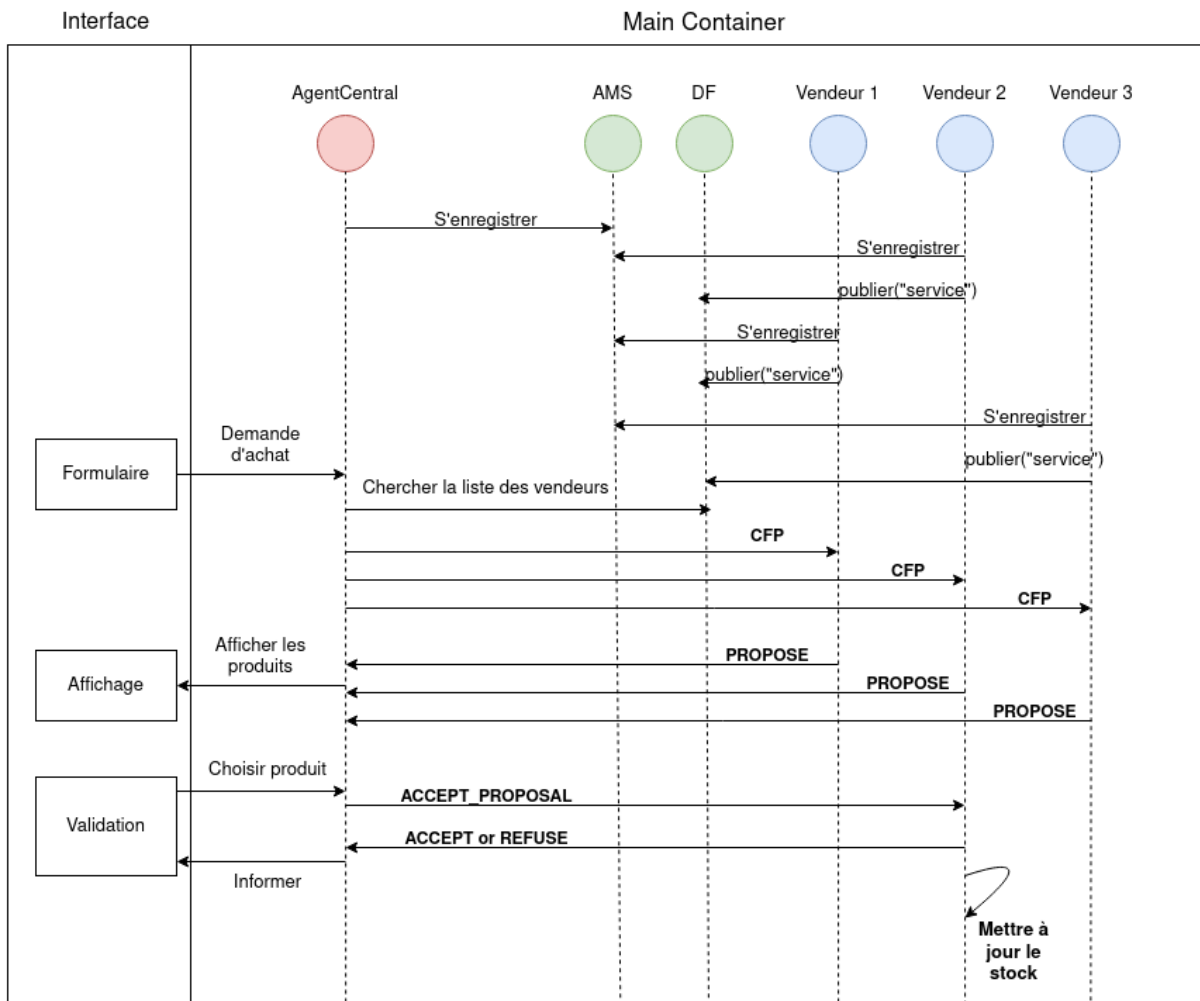
Pour la promotion, on a ajouter pour chaque agent une partie promotion dans sa base de connaissance:

```
"promotion": [  
  {  
    "id":,  
    "marque": "",  
    "taille":,  
    "quantity":,  
    "prix":  
  
    // d'autre caractéristiques_de_chaque_catégorie_promotion  
  }  
]  
}
```

Chaque agent fait appel à la fonction raisonner pour traiter la requête du client, la fonction prend en entrée les informations transmises par le client, la catégorie d'achat et la base de connaissance de chaque vendeur sous format json, et retourne la liste des produits disponibles.

Nous avons travaillé avec un seul container qui est le *MainContainer*, dans lequel nous avons créé 4 agents dont l'agent central qui s'occupe de la coordination entre les agents vendeurs et le client qui transmet sa demande via une interface graphique.

La figure ci-jointe résume l'approche choisie pour construire notre système multi-agents.



**Figure: Diagramme résumant les fonctionnalités major de notre application**

## Outils choisis pour l'implémentation

Dans cette partie du projet, nous avons utilisé la plateforme jade qui utilise le langage de programmation java.

JADE est un cadre logiciel qui facilite le développement des systèmes multi-agents sous le standard FIPA pour lequel il crée plusieurs conteneurs pour les agents, chacun d'eux pouvant s'exécuter sur un ou plusieurs systèmes. Il est entendu qu'un ensemble de conteneurs constitue une plate-forme.[3]

Nous avons aussi utilisé, comme dans la première partie du projet, le framework JavaFX pour la création de notre interfaces.

## Test

Dans cette section, nous allons faire les tests sur les différentes fonctionnalités demandées dans le projet.

- L'acheteur peut acheter un produit complet (exemple : un ordinateur) d'un agent annexe ou une ou plusieurs parties d'un produit (exemple : disque dur, mémoire, microprocesseur, ...) de plusieurs agents annexes.

Bienvenu, veuillez choisir une des catégories

RAM	PC
DISQUE DUR	CARTE GRAPHIQUE

[Voir offre de promotion](#)

- L'interface doit permettre à l'acheteur de faire entrer les critères une à une

Une marque en particulier?

- L'acheteur peut acheter un autre produit dès qu'il valide l'achat du premier.

<div><div><div>magasin3</div><div>intel</div><div>5000</div><div>Quantity <input type="text" value="3"/></div><div>Acheter</div></div><div><div>magasin2</div><div>intel</div><div>5000</div><div>Quantity <input type="text" value="3"/></div><div>Acheter</div></div></div>				
Menu Principale	<b>Vendeur1</b> rien	<b>Vendeur2</b> (PROPOSE :sender ( agent-... :receiver (set ( ... :content "[")	<b>Vendeur3</b> (PROPOSE :sender ( agent-ide. :receiver (set ( age. :content "[\\"categ.	Offre de promotion

Magasin	<input type="text" value="magasin1"/>	
Catégorie	<input type="text" value="pc"/>	
Marque	<input type="text" value="lenovo"/>	
Prix unitaire	<input type="text" value="50000"/>	X <input type="text" value="1"/>
Prix totale	<input type="text" value="50000"/>	
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>		

- Chaque agent annexe propose des réductions suivant la quantité achetée.

Prix unitaire	<input type="text" value="5000"/>	X <input type="text" value="3"/>
Prix totale	<input type="text" value="12000.0"/>	-20%

- Pour la gestion de stocks, chaque agent annexe propose des réductions.

<div> <div>magasin1</div> <div>pc lenovo</div> <div>50000</div> <div>Quantity <input type="text" value="1"/></div> <div> <input type="button" value="Acheter"/> </div> </div>	<div> <div>magasin2</div> <div>carte... nvidia</div> <div>10000</div> <div>Quantity <input type="text" value="1"/></div> <div> <input type="button" value="Acheter"/> </div> </div>	<div> <div>magasin3</div> <div>ram sHynix</div> <div>7000</div> <div>Quantity <input type="text" value="1"/></div> <div> <input type="button" value="Acheter"/> </div> </div>
<div> <div>magasin3</div> <div>Menu Principale</div> </div>		<div> <div>Offre de promotion</div> </div>

- Toutes les informations échangées entre l'agent central et les agents annexes doivent être affichées sur l'interface.

Vendeur1	Vendeur2	Vendeur3
rien	(PROPOSE :sender ( agent-... :receiver (set ( ... :content "[ ]"	(PROPOSE :sender ( agent-ide. :receiver (set ( age. :content "[{"categ.

## **Conclusion Générale**

Dans ce projet, nous avons eu l'occasion de concevoir et implémenter un système expert et multiagent à l'aide de la plateforme jade, et cela nous a permis de nous familiariser avec un des concepts les plus importants de l'intelligence artificielle à savoir les systèmes multi agents.

## **Référence**

- [1]: <https://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501897-syste-me-expert-definition-fonctionnement-et-exemples/>
- [2]: [https://fr.wikipedia.org/wiki/Syst%C3%A8me\\_multi-agents](https://fr.wikipedia.org/wiki/Syst%C3%A8me_multi-agents)
- [3]: [https://en.wikipedia.org/wiki/Java\\_Agent\\_Development\\_Framework](https://en.wikipedia.org/wiki/Java_Agent_Development_Framework)