

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université des Sciences et de la Technologie Houari Boumédiène

Faculté d'Informatique

Département d'Intelligence Artificielle et Sciences de Données

Master Systèmes Informatiques intelligents

Module : Technologie des agents

Rapport de projet de TP

Réalisé par :

BOUROUINA Rania, 181831052716

CHIBANE Ilies, 181831072041

Année universitaire : 2021 / 2022

Table des matières

1	Introduction	2
2	Implémentation d'un système expert	3
2.1	Principe.....	3
2.2	Explication du système expert	3
2.3	Explication de l'interface graphique et de l'API	5
2.4	Base de connaissance utilisée.....	6
2.4.1	Base de connaissance véhicule	6
2.4.2	Base de connaissance produits électroniques	6
3	Implémentation d'un environnement commercial à base d'agents intelligents sous forme d'un système expert	8
3.1	Principe.....	8
3.2	Explication du système multi-agent	8
3.3	Explication de l'interface graphique et de l'API	9
4	Introduction	10

Chapitre 1

Introduction

Les agents intelligents sont une notion essentielle en intelligence artificielle, utilisé dans de nombreux domaines internes et hors informatique, il est considéré comme un paradigme important pour tout informaticien, même en dehors du domaine de l'IA. C'est pourquoi dans ce projet, nous mettons en avant les agents intelligents sous forme d'un système expert dans un environnement commercial, le tout sous le format d'API REST. Les technologies web étant de plus en plus populaire et utilisé dans de nombreux domaines, il est particulièrement intéressant de combiner ceci avec une technologie aussi polyvalente que les systèmes multi-agents. Ce projet a été réalisé avec le langage Python ainsi que les bibliothèques SPADE pour coder les agents intelligents et le framework flask pour créer l'API REST l'interface graphique, elle a été réalisée avec le langage JavaScript en utilisant le framework Vue JS.

Chapitre 2

Implémentation d'un système expert

2.1 Principe

Pour la première partie de ce projet, nous avons développé un raisonneur capable d'exécuter un set de règles sur un système expert.

2.2 Explication du système expert

Pour ce projet, nous avons réalisé un système expert dynamique qui utilise des fichiers JSON afin d'accéder à sa base de règles, qui implique que notre système expert peut s'adapter à n'importe quel contexte tant qu'il lui est fourni le fichier JSON avec la base de règles approprié.

Dans notre cas, nous avons mis à dispositions 2 fichiers JSON, l'un contenant une base de règles pour les véhicules et un autre fichier pour diverses catégories de produits électronique tel que des smartphones, des laptops, des pc de bureau, des casques, ou même des composants électroniques. Tout ce que notre système expert a à faire est de stocker l'ensemble de ces règles dans une liste qu'il se mettra à parcourir afin de vérifier si les informations que l'utilisateur fournira en entrée satisfont certaines, règles, ce qui par conséquent aura pour effet de retourner un type de voiture ou une catégorie de produit électronique dépendant de la base de règle sélectionner.

Projet Tech Agent

Select facts

Category

Type

Size

Function

Portable

REASON

Figure 2.1 – Interface de sélection des facts d'un produit

Projet Tech Agent

Select facts

Motors

Size

Vehicule type

Number of wheels

Number of doors

REASON

Figure 2.2 – Interface de sélection des facts d'un véhicule

2.3 Explication de l'interface graphique et de l'API

En ce qui concerne l'interface graphique de notre projet, nous avons opté comme cité précédemment pour une interface web qui est divisée en 4 fenêtres, la première permettant à l'utilisateur de rentrer les informations du produit électronique qu'il recherche et la seconde permettant de réaliser la même chose, mais pour les véhicules.

Une fois que l'utilisateur appuie sur le bouton **Reason** un message est envoyé à l'API dépendant de la page sur laquelle l'utilisateur se trouvait afin de déterminer quelle base de connaissance le système expert doit utiliser et une fois cela fait il retourne le résultat obtenu et l'affiche sur l'interface.

Project Tech Agent

Select facts

Computers Device

Doesn't matter Sender

Doesn't matter

REASON

List of places to visit
Laptop
desktop computer

Figure 2.3 – Résultat obtenu par le système expert en utilisant la base de connaissance « produits »

Project Tech Agent

Select facts

no small

cycle Doesn't matter Doesn't matter

REASON

List of vehicle
Bicycle
Tricycle

Figure 2.4 – Résultat obtenu par le système expert en utilisant la base de connaissance « véhicules »

2.4 Base de connaissance utilisée

2.4.1 Base de connaissance véhicule

1. IF Vehicule_Type = “cycle” AND Number_of_wheels = “2” AND Motor = "No" THEN Vehicle = “Bicycle”
2. IF Vehicule_Type = “cycle” AND Number_of_wheels = “3” AND Motor = "No" THEN Vehicle = “Tricycle”
3. IF Vehicule_Type = “cycle” AND Number_of_wheels = “2” AND Motor = "Yes" THEN Vehicle = “Motorcycle”
4. IF Vehicule_Type = “automobile” and Size = “Small” and Number_of_doors = ”4” THEN Vehicle = “Sports Car”
5. IF Vehicule_Type = “automobile” and Size = “Medium” and Number_of_doors = ”4” THEN Vehicle = “Sedan”
6. IF Vehicule_Type = “automobile” and Size = “Medium” and Number_of_doors = ”3” THEN Vehicle = “MiniVan”
7. IF Vehicule_Type = “automobile” and Size = “Large” and Number_of_doors = ”4” THEN Vehicle = “SUV”
8. IF Number_of_wheels < “4” THEN Vehicule_Type = “Cycle”
9. IF Number_of_wheels = “4” THEN Vehicule_Type = “Automobile”

2.4.2 Base de connaissance produits électroniques

1. IF Category = “Audio” AND Type = “Device” AND Function = "Receiver" THEN produit_a_acheter = “Headphones”
2. IF Category = “Audio” AND Type = “Device” AND Function = "Sender" THEN produit_a_acheter = “Microphone”
3. IF Category = “Video” AND Type = “Component” AND Function = "Receiver" THEN produit_a_acheter = “Afficheur”
4. IF Category = “Computers” AND Type = “Device” AND Portable = "Yes" THEN produit_a_acheter = “Laptop”

5. IF Category = “Electronics” AND Type = “Component” AND Size = "Small" THEN produit_a_acheter = “Semiconducteur”
6. IF Category = “Video” AND Type = “Component” AND Portable = "No" AND Function = "Receiver" THEN produit_a_acheter = “TV Screen”
7. IF Category = “Video” AND Type = “Device” AND Portable = "No" AND Function = "Receiver" THEN produit_a_acheter = “Oculus”
8. IF Category = “Computers” AND Type = “Device” AND Portable = "No" THEN produit_a_acheter = “desktop computer”
9. IF Category = “Electronics” AND Type = “Component” AND Function = "Sender" THEN produit_a_acheter = “RAM”
10. IF Category = “Electronics” AND Type = “Component” AND Function = "Receiver" THEN produit_a_acheter = “SSD”
11. IF Category = “Mobile” AND Type = “Device” AND Size = "Small" THEN produit_a_acheter = “Smartphone”
12. IF Category = “Mobile” AND Type = “Device” AND Size = "Medium" THEN produit_a_acheter = “Tablet”
13. IF Category = “Audio” AND Type = “Device” THEN Portable = “Yes”
14. IF Category = “Computers” THEN Function = “Sender”
15. IF Category = “Electronics” AND Type = “Component” THEN Size = “Small”

Il est possible de vérifier les différentes dans la page rules du site ils sont affichés grâce à un message envoyé à l’API qui renvoient les différentes règles.

Projet Tech Agent	
Rules	
List of vehicles facts	List of products facts
Rule 1 : IF Vehicule_Type == cycle AND IF Number_of_wheels == 2 AND IF Motor == no THEN Bicycle Rule 2 : IF Vehicule_Type == cycle AND IF Number_of_wheels == 3 AND IF Motor == no THEN Tricycle Rule 3 :	Rule 1 : IF Category == Audio AND IF Type == Device AND IF Function == Receiver THEN Headphones Rule 10 : IF Category == Electronics AND IF Type == Component AND IF Function == Sender THEN RAM Rule 11 :

Figure 2.5 – Listes des règles du système expert

Chapitre 3

Implémentation d'un environnement commercial à base d'agents intelligents sous forme d'un système expert

3.1 Principe

Pour cette deuxième partie de ce projet, nous avons développé un système multi-agent reliant un agent principal communiquant avec 3 agents annexes utilisant le système expert développé lors de la partie 1.

3.2 Explication du système multi-agent

Comme expliqué juste avant, notre système multi-agent est composé d'un agent principal ainsi que de 3 agents annexes, réaliser avec la librairie **SPADE (Smart Python Agent Development Environment)** .

Nos agents ont 2 états principaux, un état "Sender" et un autre état "Receiver", lorsqu'un agent est dans l'état Sender il envoie un message à un agent dans l'état Receiver une fois son action terminée il passe à l'état Receiver et vice versa, cependant certaines fonctionnalités des agents ne dépendent pas de leurs états et seront plus détaillées par la suite.

Notre agent principal, une fois avoir reçu les informations de la part de l'utilisateur, les sauvegarde dans un fichier JSON qu'il envoie par la suite aux 3 agents annexes. une fois le fichier JSON reçu par les agents, ils utilisent leur système expert afin de consulter leurs bases de connaissances respectif et de trouver les produits correspondant aux critères reçu dans le fichier JSON envoyé par l'agent principal.

Une fois cela fait, chaque agent sauvegarde le résultat obtenu dans un fichier JSON qui sont par la suite envoyé à l'agent principal.

L'agent principal se charge par la suite de réorganiser le contenu des différents objets reçu en un seul et même objet JSON puis l'envoie à l'interface pour qu'il soit affiché.

Par la suite, l'utilisateur peut décider d'acheter un produit et une fois fait, l'agent principal reçoit un message notifiant l'achat et se charge de communiquer un fichier JSON contenant les détails de la transaction à l'agent gérant le magasin, possédant produit acheté et met à jour sa base de connaissance.

3.3 Explication de l'interface graphique et de l'API

L'interface marche de la même manière que lors de la partie 1 cependant l'utilité de la REST API utilisée est plus importante. En effet, elle permet de sécuriser et de réguler les différents appels faits de l'utilisateur vers l'agent principal, mais aussi de tracer plus facilement les différents échanges entre les agents, permettant par la suite d'afficher les historiques des informations échangées sur l'interface. Cela a aussi comme effet de rendre notre système plus souple, car il peut facilement et être adapté dans d'autres environnements.

The screenshot shows the 'Projet Tech Agent' interface. At the top, there is a header bar with the text 'Projet Tech Agent'. Below this, there is a section titled 'Select facts'. This section contains seven dropdown menus arranged in two rows: 'Type', 'Quantity', 'Price', 'Color', 'Promotion', 'Brand', and 'State'. Below these dropdowns, there is a button labeled 'REASON'.

Figure 3.1 – Interface de sélection d'un produit

The screenshot shows the 'Projet Tech Agent' interface with more content. At the top, there is a header bar with the text 'Projet Tech Agent'. Below this, there is a section titled 'List of product to buy'. This section contains two product cards. Each card has a title 'My Product', a list of attributes (Color, Size, Weight, Price, Promotion, Brand, State), and a 'Buy' button. To the right of the product cards, there is a section titled 'agents communication history' which displays a list of messages exchanged between agents.

Figure 3.1 – Résultat obtenu par le système multi-agent

Chapitre 4

Conclusion

Ce projet nous aura permis d'expérimenter l'utilité et le fonctionnement des systèmes experts et multi-agents, démontrant leur efficacité et leurs flexibilités afin de résoudre des problèmes dans différents domaines comme celui-ci lié à l'environnement commercial ou les différents agents reliés entre eux ont permis avec les utilisations de leurs différentes bases de connaissances à proposer le meilleur produit répondant aux exigences de notre utilisateur.

Et en bonus, nous avons pu constater la bonne chimie que les systèmes multi-agents ont avec les REST API prouvant qu'il s'agit d'un domaine intéressant à exploiter et à améliorer.