

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



Master Systèmes Informatiques intelligents

Module : Technologie des agents

RAPPORT DES PARTIES 1 & 2 DU PROJET

Systèmes Experts et Multi Agents

Réalisé par :

TOLBA Yasmine

TOUMACHE Doudja Rania

2021/2022

Introduction:	3
Partie 1: Les systèmes experts	3
Les règles du premier système	3
Les règles du deuxième système	3
Fonctionnement	4
Partie 2: Le système multi agents	5
La technologie multi-agents	5
En code python	5
La communication entre les agents et leurs états	5
Schéma de communication entre les agents et de leurs fonctionnement	6
Conclusion	9
Annexes	10
Partie 1: Système expert	10
Partie 2: Système multi agents	12

Introduction:

De nos jours, l'intelligence artificielle est répandue et utilisée dans plusieurs domaines. On retrouve des ordinateurs et machines pouvant imiter et réaliser des performances qui sont habituellement exécutées par des êtres humains. Parmi les multiples succès de l'IA, on peut citer les systèmes experts et systèmes multi agents (SMA).

Un système expert est un outil qui nous permet de reproduire le comportement d'un expert humain commettant une certaine tâche intellectuelle dans une spécialité précise. Ces systèmes sont généralement utilisés pour résoudre des problèmes de classification ou de décision. Un système expert est composé d'une base de connaissance et d'un moteur d'inférence. Notre base de connaissance sera constituée de différents faits et règles, alors que le moteur d'inférence représentera le mécanisme de raisonnement.

Un système multi agents nous offre une approche adaptée pour traiter des problèmes complexes. Ce système est constitué de plusieurs agents intelligents qui interagissent et communiquent entre eux pour résoudre des problèmes de manière autonome.

Dans ce projet, nous allons développer une application desktop codée en langage python. Nous allons utiliser le module Experta pour construire deux systèmes experts distincts, et en utilisant la bibliothèque Spade, ces derniers seront reliés à un système multi agents simulant un système de vente et achat où l'utilisateur pourra rechercher des produits en saisissant les caractéristiques de ces derniers.

1. Partie 1: Les systèmes experts

Dans la première partie de ce projet, on va implémenter un système expert dans le domaine commercial.

Nous avons conçu deux systèmes experts, le premier est un système pour reconnaître les types des véhicules, le deuxième est un système pour reconnaître les types et modèles de réfrigérateurs.

1.1. Les règles du premier système

- Si moins de 4 roues: type = Cycle
- Si 4 roues et avec moteur: type = Automobile
- Si type = Cycle et sans moteur et 2 roues: véhicule = bicyclette
- Si type = Cycle et sans moteur et 3 roues: véhicule = tricycle
- Si type = Cycle et avec moteur et 3 roues: véhicule = Moto
- Si type = Automobile et 2 portes et petite taille: véhicule = voiture de sport
- Si type = Automobile et 4 portes et taille moyenne: véhicule = Sedan
- Si type = Automobile et 3 portes et taille moyenne: véhicule = Mini van
- Si type = Automobile et 4 portes et grande taille: véhicule = SUV

1.2. Les règles du deuxième système

- Si nombre de portes supérieur ou égal à 3 : portes = portes françaises
- Si plus haute température inférieure ou égale à -18°C : type = congélateur
- Si plus haute température supérieure à -18°C : type = réfrigérateur

- Si plus basse température inférieure ou égale à -18°C : réfrigérateur = a un congélateur
- Si plus basse température supérieure à -18°C : réfrigérateur = n'a pas de congélateur
- Si type = congélateur et grande taille : type = congélateur coffre
- Si type = congélateur et taille petite : type = congélateur armoire
- Si type = réfrigérateur et taille petite et 1 porte et n'a pas de congélateur = mini réfrigérateur
- Si type = réfrigérateur et grande taille et 2 portes et a un congélateur : réfrigérateur = côte à côte
- Si type = réfrigérateur et grande taille et portes françaises et a un congélateur : réfrigérateur = réfrigérateur à portes françaises
- Si type = réfrigérateur et taille moyenne et 2 portes et a un congélateur : réfrigérateur = réfrigérateur standard
- Si type = réfrigérateur et taille moyenne et 2 portes et n'a pas de congélateur : réfrigérateur = réfrigérateur simple
- Si type = réfrigérateur et grande taille et portes françaises et a un congélateur et muni d'une tablette : réfrigérateur = réfrigérateur intelligent
- Si type = réfrigérateur et très grande taille et 2 portes et n'a pas de congélateur : réfrigérateur = réfrigérateur colonne
- Si distributeur d'eau : réfrigérateur muni d'un distributeur d'eau
- Si distributeur de glaçons : réfrigérateur muni d'un distributeur de glaçon
- Si tablette : réfrigérateur muni d'une tablette

1.3. Fonctionnement

Notre moteur d'inférence contiendra donc les règles des 2 systèmes précédemment mentionnés, et pour cela nous allons utiliser le module Experta de Python.

Grâce à ce module, tout ce que nous avons à faire est de déclarer notre moteur d'inférence, et de déclarer les règles, ainsi que les actions à faire lorsque les règles sont déclenchées.

Certaines règles doivent faire appel à d'autres règles, pour cela, nous devons déclarer des règles qui ajoutent leur déclenchement à la liste des faits. En faisant cela, les règles qui doivent faire appel à d'autres règles auront le déclenchement de ces dernières en tant que faits dans la liste des faits déclarés dans le moteur.

Pour lancer le moteur d'inférence et recevoir des résultats de notre système expert, nous devons déclarer des faits, ces derniers sont par exemple: le nombre de roues, le nombre de portes, avec ou sans moteur, etc...

Certaines règles ont pour unique but être appelés par d'autres règles, pour remédier à cela, le déclenchement de ces dernières aura pour unique action, d'ajouter un fait au moteur d'inférence, par exemple, si le nombre de roues est inférieur à 4, alors le type sera cycle. Ce qui veut dire que si nous déclarons le fait qu'il y a 2 roues, la règle ajoutera automatiquement au moteur, le fait que le type soit un cycle.

Une fois les faits déclarés et le moteur déclenché, ce dernier déclenche automatiquement les règles qui répondent aux faits donnés, et dans les actions des règles, nous retournerons le résultat.

2. Partie 2: Le système multi agents

Dans cette partie, nous allons évoluer le système développé en partie 1, et cela en utilisant la technologie multi-agents, pour réaliser un système de recommandation et de commerce en 1.

2.1. La technologie multi-agents

en utilisant le module Spade de python, on pourra réaliser un système où l'utilisateur donnera en entrée les spécifications qu'il souhaite, et aura en retour un ensemble de produits recommandés à acheter. Pour cela, la technologie multi agents sera utile. Un agent principal représente comme une personne, ou un agent commercial qui communique avec le client et les magasins, chaque magasin est géré par des agents annexes, qui communiquent uniquement avec l'agent principal. La communication entre ces agents se fait comme suit: l'agent principal reçoit les préférences de l'utilisateur, il envoie un message aux agents annexes leur communiquant ce que l'utilisateur recherche, les agents annexes fouillent dans leur base de données, et renvoient à l'agent principale tous les produits qui correspondent aux souhaits de l'utilisateur. L'agent principal affiche à l'utilisateur les résultats obtenus, ce dernier aura le choix d'acheter les articles qu'il souhaite, en les mettant dans un panier, puis en validant ses achats. Une fois les achats validés, l'agent principal communique cette information aux agents annexes, pour mettre à jour leur base de données et de leurs stocks.

2.2. En code python

Pour programmer ceci en python, le module Spade a été utilisé. Sur Spade, un agent a besoin d'un "comportement" behavior. Ce comportement aura pour tâche de définir la manière dont l'agent s'exécute, s'il s'exécute une fois, se répète plusieurs fois, ou de changer d'état. Vu le comportement de nos agents que nous avons mentionné plus tôt, le meilleur "behavior" est celui basé sur les états, le "FSMBehavior". FSM veut dire Final State Machine, dont une machine à état fini. Ce comportement est programmé tout comme un automate d'états, la raison pour laquelle nous avons choisi ce comportement est car les agents changent d'état, selon l'étape dans laquelle ils se trouvent. En effet, les agents changent plusieurs fois entre l'état d'envoi de message, et d'attente d'un message, on ne peut pas attendre et envoyer un message en même temps, donc on sépare les deux et on change selon les besoins.

2.3. La communication entre les agents et leurs états

l'agent principal aura comme état initial l'envoi de message, tandis que les agents annexes auront l'attente de message comme état initial. Cela est tout simplement dû au fait que le déroulement des événements commence par l'envoi des souhaits de

l'utilisateur aux agents annexes, c'est une fois le premier message envoyé et reçu que les agents changent d'état.

2.4. Schéma de communication entre les agents et de leurs fonctionnement

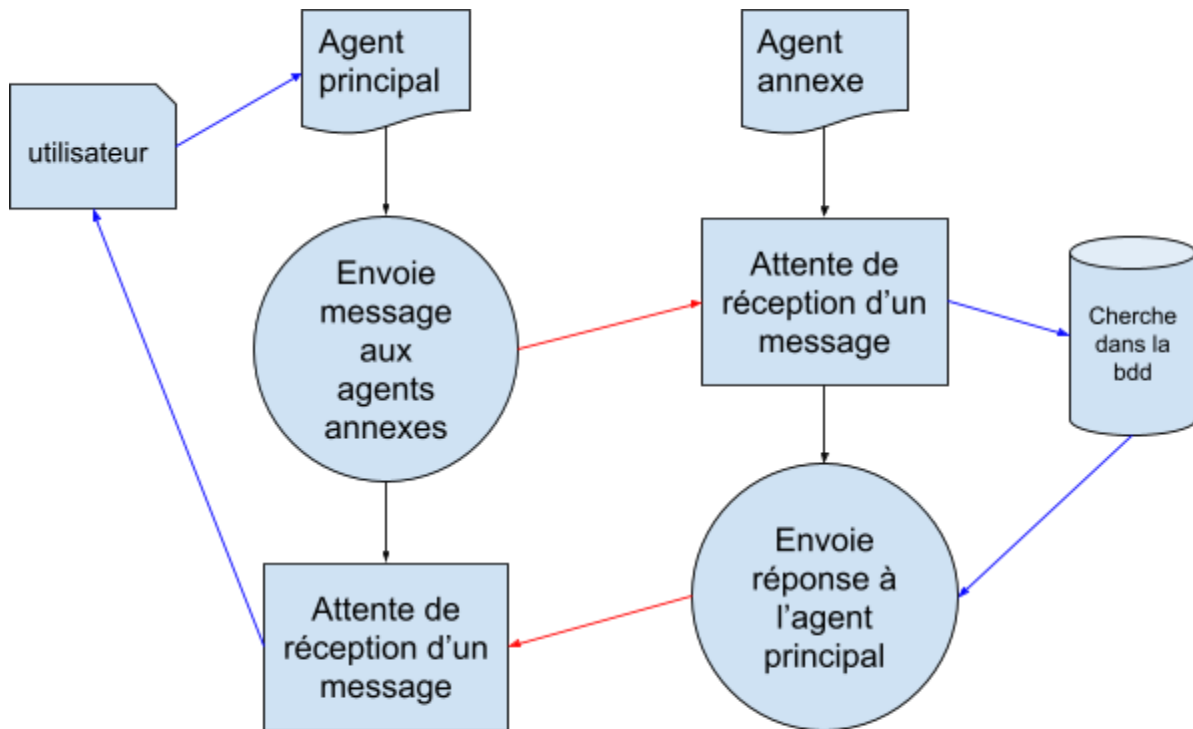


Figure 1: Diagramme du fonctionnement général du système et de la communication multi-agents

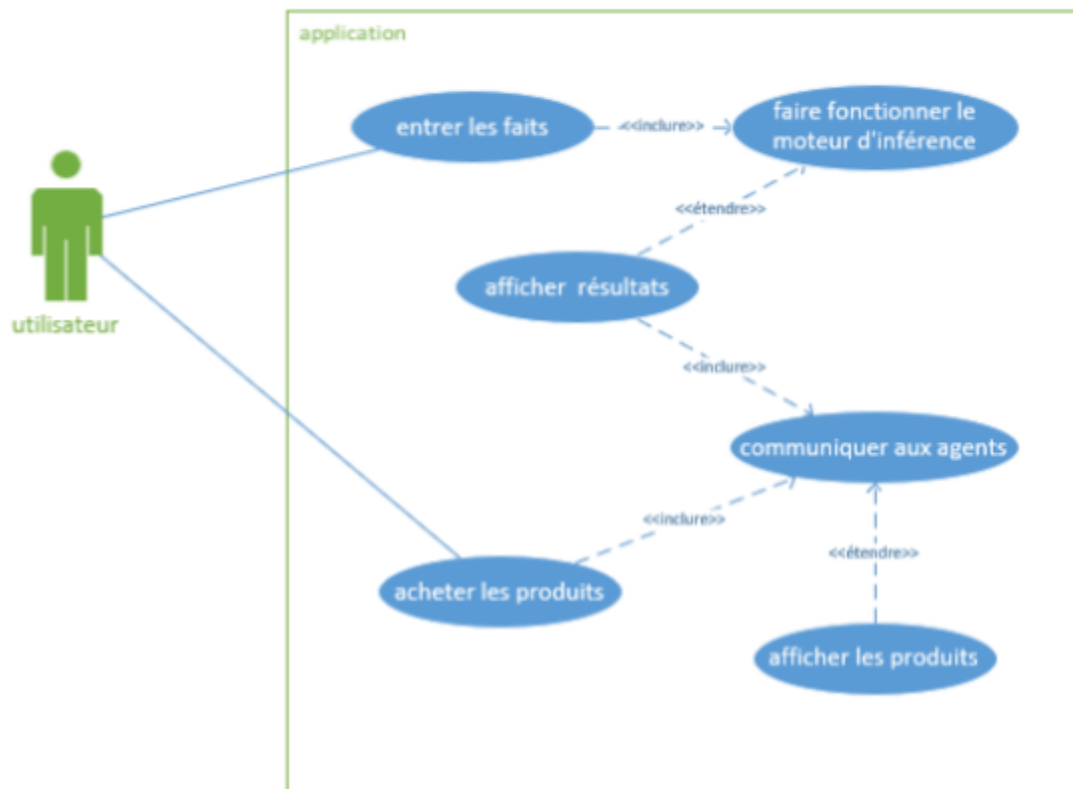


Figure 2: Diagramme cas-utilisation entre l'utilisateur et l'application

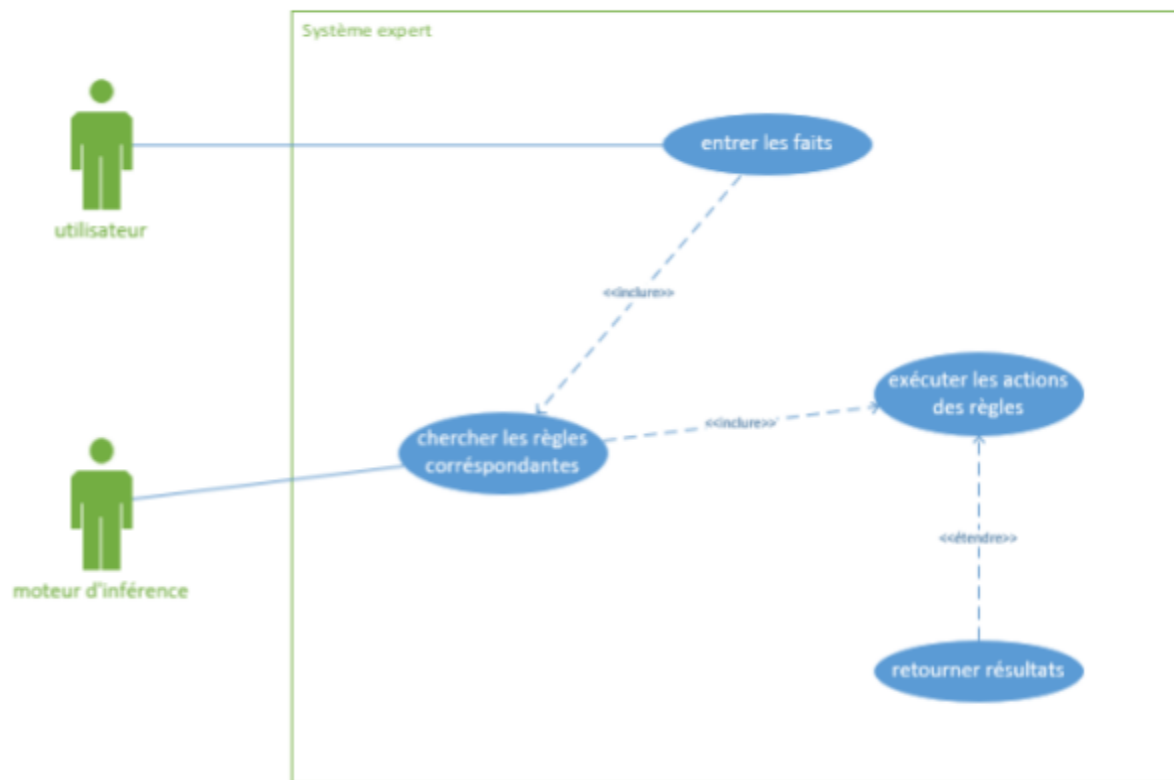


Figure 3: Diagramme cas-utilisation du système expert

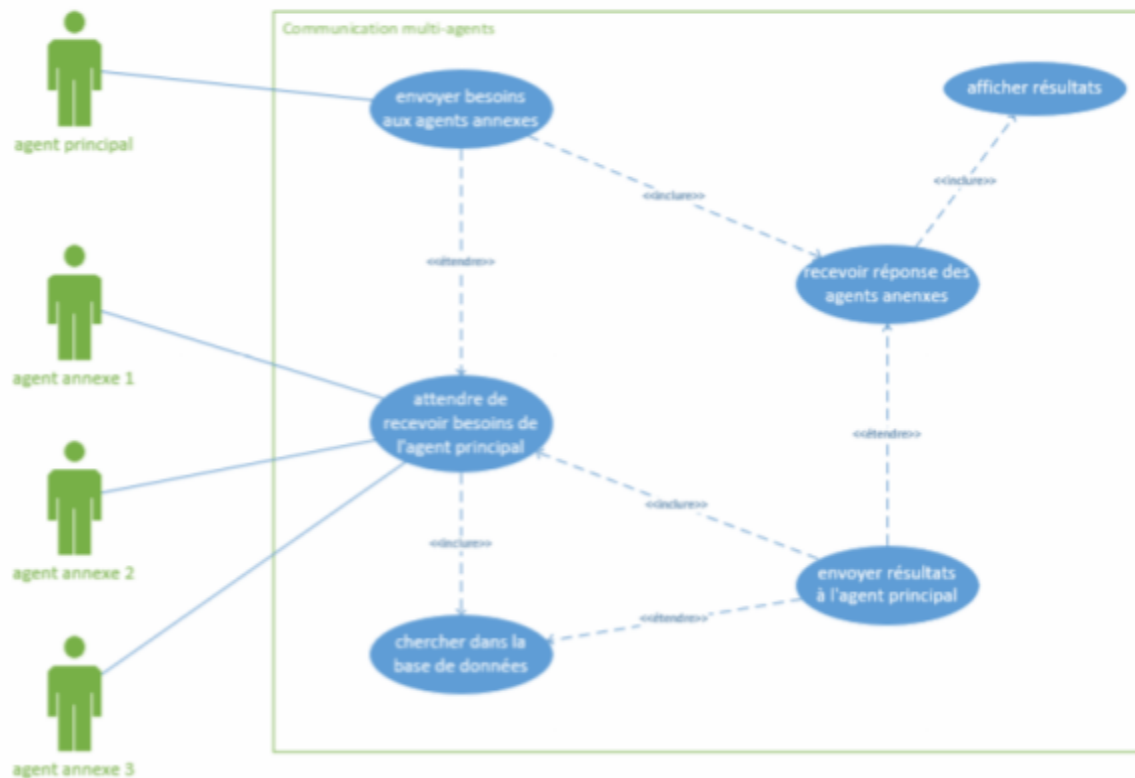


Figure 4: Diagramme cas-utilisation de la communication multi-agents

3. Conclusion

sur ce projet nous avons pu étudier en profondeur le fonctionnement et surtout l'application des systèmes experts et des agents intelligents dans des conditions réelles, dans le domaine commercial. On peut donc constater l'efficacité et l'étendue des capacités de ces ressources, et envisager de l'utiliser dans plusieurs domaines différents, comme les diagnostics médicaux, la gestion des robots intelligents, les systèmes de jeux, ...etc. Les systèmes experts nous permettent de déterminer des recherches rapides en utilisant des notions de logique simple, avec la possibilité d'optimiser le code, même avec des règles complexes. Les agents intelligents sont une ressource très importante et intéressante dans les systèmes composés, et cela est dû à sa sécurité, et la possibilité des agents de communiquer entre eux, ainsi d'interagir avec leur environnement, tout cela en simulant une vraie personne avec une intelligence, ou bien de faire fonctionner plusieurs processus dans une application en même temps d'une manière intelligente et complexe.

4. Annexes

- Partie I: Système expert

- A l'ouverture de l'application:

Vehicles

Refrigerators

Medical Diagnosis

number of wheels

2

+

number of doors

0

+

motor

yes

+

size

Small

+

List of facts

remove

get vehicle

reset

Your Vehide(s)

Invocated Rules

- Exemple de véhicule:

Vehicles
Refrigerators
Medical Diagnosis

number of wheels

4

+

number of doors

4

+

motor

yes

+

size

Medium

+

List of facts

wheels=4
motor=yes
doors=4
size=Medium

remove

get vehicle

reset

Your Vehicle(s)

Sedan

Invoked Rules

if doors = 4 and size = Medium and type = a

- Exemple de réfrigérateur:

Vehicles
Refrigerators
Medical Diagnosis

number of doors

1

+

size

Small

+

highest temperature

1,50

C°

+

lowest temperature

0,00

C°

+

additional features

Tablet

+

List of facts

f_doors=1
size=Small
hi_temp=1.5
lo_temp=0.0

remove

get fridge

reset

Fridge

Mini Fridge

Invoked Rules

if high temperature > -18 then type = fridge
if low temperature > -18 then does not have
if size = small and doors = 1 and does not ha

- Partie 2: Système multi agents

- À l'ouverture de l'application

The screenshot shows a window titled "multi agents" with a light blue background. It contains several sections for product search and management:

- Filters:** On the left, there are four filter groups, each with a dropdown menu and a "+" button:
 - Size: Small
 - Doors: 1
 - Temperature Max: 1,50
 - Temperature Min: -18,00
- Input Facts:** A large empty text area with a "remove" button to its right.
- Search:** A "search" button is located below the input facts area.
- Reset:** A "reset" button is located below the search button.
- Search Results:** A large empty text area at the bottom left.
- Recommendations:** A table with columns ID, Name, color, and price. It is currently empty.
- Add to cart:** A button located below the recommendations table.
- Cart:** A table with columns ID, Name, color, and price. It is currently empty.
- Cart Actions:** Below the cart table are three buttons: "Purchase", "remove", and "reset".

- Exemple d'achat: recherche d'un produit

multi agents

Size

Small

+

Doors

1

+

Temperature Max

1,50

+

Temperature Min

0,00

+

Input Facts

size= Small

doors = 1

hi_temp = 1.5

lo_temp = 0.0

accessories=ice maker

remove

search

reset

Search Results

Mini Fridge

Features

Tablet

+

Accessories

ice maker

+

Recommendations

	ID	Name	color	
1	6	Hisense 45L Bar...	White	3626
2	7	Hisense 125L B...	Black	5083
3	8	Hisense 125L B...	White	5083
4	3	Hisense 125L B...	Black	5083
5	8	Frigidaire ...	NA	1019

Add to cart

Cart

ID	Name	color	price
----	------	-------	-------

Purchase

remove

reset

- Exemple d'achat : sélection de produits

multi agents

Size

Small

+

Doors

1

+

Temperature Max

1,50

+

Temperature Min

0,00

+

Input Facts

size= Small

doors = 1

hi_temp = 1.5

lo_temp = 0.0

accessories=ice maker

remove

search

reset

Search Results

Mini Fridge

Features

Tablet

+

Accessories

ice maker

+

Recommendations

	ID	Name	color	
2	7	Hisense 125L B...	Black	5083
3	8	Hisense 125L B...	White	5083
4	3	Hisense 125L B...	Black	5083
5	8	Frigidaire ...	NA	1019
6	9	GE WR30X1009...	White	2170

Add to cart

Cart

ID	Name	color	p
1	7	Hisense 125L B...	50833
2	8	Hisense 125L B...	50833
3	8	Frigidaire ...	10195

Purchase

remove

reset

- Exemple achat : acheter produit

multi agents

Size:

Doors:

Temperature Max:

Temperature Min:

Features:

Accessories:

Input Facts

size=Small
doors = 1
hi_temp = 1.5
lo_temp = 0.0
accessories=ice maker

Search Results

Mini Fridge

Dialog

Receipt

ID	Name	color	price
1 7	Hisense 125L B...	Black	50833
2 8	Hisense 125L B...	White	50833
3 8	Frigidaire ...	NA	10195

Total:

Purchase

color

color	price
Black	5083
White	5083
Black	5083
NA	1019
White	2170

multi agents

Size:

Doors:

Temperature Max:

Temperature Min:

Features:

Accessories:

Input Facts

size=Small
doors = 1
hi_temp = 1.5
lo_temp = 0.0
accessories=ice maker

Search Results

Mini Fridge

Dialog

Receipt

ID	Name	color	price
1 7	Hisense 125L B...	Black	50833
2 8	Hisense 125L B...	White	50833
3 8	Frigidaire ...	NA	10195

Succès

Achat effectué avec succès

Total:

Purchase

color

color	price
Black	5083
White	5083
Black	5083
NA	1019
White	2170