



Technologie des agents

Rapport du mini projet :

« Implémentation d'un processus d'enchère avec un système multi-agent »

Spécialité : Master 1 SII

Binôme :

**CHIKH Khadidja
SAHARI Hadjer**

24/07/2019

Contexte :

Dans le cadre du module “technologie des agents”, il nous a été demandé de:

- Implémenter un système expert adapté au domaine médical (sous Java).
- Implémenter des agents intelligents pour simuler un processus d'enchère, sous la plateforme JADE(sous Java).
- Refaire la deuxième partie sous la plateforme JASON et établir une comparaison entre les deux plateformes.

Partie 1:

Dans cette partie, nous avons implémenté un système expert (application java) de vehicules en se basant sur le livre « *Constructing Intelligents with java* [1]» (chapitres 3 et 4). Ce système est basé sur le raisonnement par règles.

Système à base de règles (rule based system) :

En informatique, un système à base de règles est un ensemble d'instructions "if-then" qui utilisent un ensemble d'assertions, auxquelles sont définies des règles sur la manière de traiter ces assertions. En développement logiciel, les systèmes à base de règles peuvent être utilisés pour créer un logiciel qui apportera une réponse à un problème à la place d'un expert humain. Ce type de système peut également être appelé système expert. Les systèmes à base de règles sont également utilisés dans la programmation et les systèmes d'intelligence artificielle. [2]

Nous avons ensuite adapté notre système au domaine médical, et ceci en introduisant de nouvelles règles inspirées des cas réels.

Exemple : Hypertension : If pressionalterielle>13 then hypertension=Oui

AVC : If hypertension=Oui
and paralysiehemicorps=Oui
then AVC=Oui

Interface graphique de l'application :

Partie de l'expert (medecin) :

A travers l'interface, l'utilisateur peut selectionner une base de règle (unique dans notre cas), voir les règles existantes dans cette base, selectionner des symptomes et des valeurs à ceux-ci, et executer. Le résultat est affiché par la suite.

Dans la figure suivante nous pouvons voir le résultat obtenu : Hypertension=Oui et le diagnostic= AVC pour les symptomes : paralysiehemicorps=Oui et pressionalterielle=14.

Visualiseur d'applets : rule.RuleApplet.class

Applet

Rule Base: Malade

Variable: Paralysie hemicorps Value: Oui

Intoxication: IF Fièvre=Oui
AND Diarrhée=Oui
AND Vomissement=Oui
THEN Diagnostic=Intoxication
Rhume: IF Maux de tête=Oui
AND Rhinorrhée=Oui
THEN Diagnostic=Rhume

Ventre de bois value = null
Urines foncées value = null
Ictère value = null
Pression artérielle set to 14
Paralysie hemicorps set to Non
Paralysie hemicorps set to Oui

Trace Log

Hypertension value = Oui
Rhinorrhée value = null
Diarrhée value = null
Courbatures value = null
Selles décolorées value = null
Douleur abdominale value = null
Fièvre value = null
Maux de tête value = null
Diagnostic value = AVC
Vomissement value = null

☒ Forward Chain
☐ Backward Chain

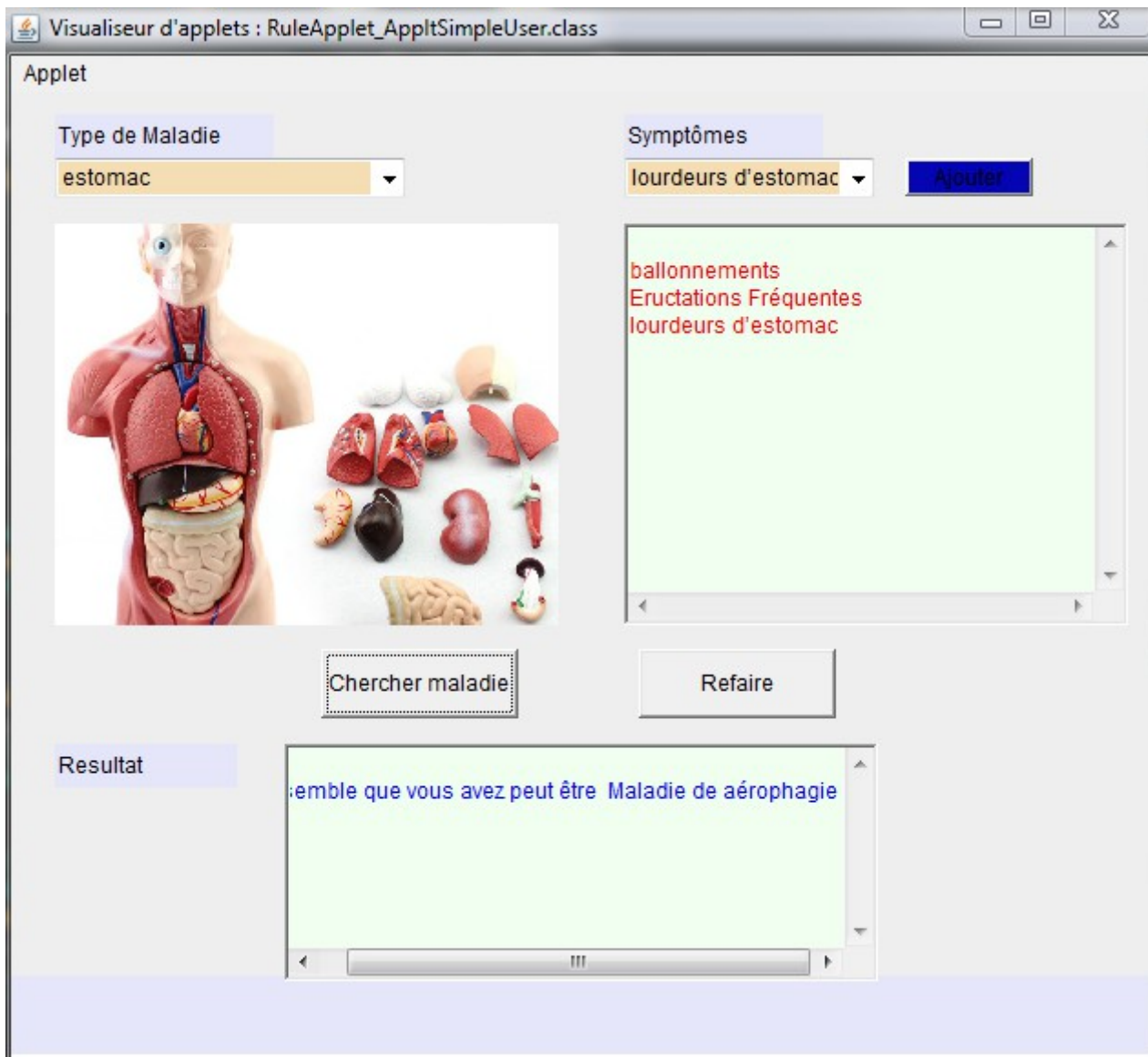
Goal

Find Goal Reset

Applet démarrée.

Partie d'un utilisateur normal(malade) :

A travers l'interface, l'utilisateur peut sélectionner un type de maladie ensuite ajouter les symptômes qu'il a, et exécuter. Il obtient ainsi un diagnostic possible pour son état (comme le montre la figure suivante).



Partie 2 :

Dans cette partie, nous avons développé une application qui simule une enchère, qui est un bon exemple de la negociation compétitive, en utilisant les agents intelligents sous JADE, représentant le vendeur et les acheteurs.

Agent:

Un agent est essentiellement un composant logiciel spécial qui a une autonomie qui fournit une interface interopérable avec un système arbitraire et / ou se comporte comme un agent humain, travaillant pour certains clients dans la poursuite de son propre agenda.[3]

Système multi-agent :

La communication de systèmes multi-agents est une technologie qui permet à plusieurs agents intelligents en interaction de communiquer entre eux et avec l'environnement.[4]

JADE :

JADE est un framework dédié au développement des applications multi-agents. Il est entièrement implémenté en Java et totalement Open Source. JADE comprend un conteneur d'agents dans chaque hôte pour maintenir ses agents locaux. Chaque agent est un *thread* actif qui a ses propres comportements. Les agents dans JADE sont exécutés simultanément. Il fournit une plate-forme d'agent virtuelle, par laquelle tous les agents peuvent interagir les uns avec les autres, indépendamment de leurs conteneurs ou hôtes.[5]

Processus de l'enchère :

Le processus de l'enchère se présente ainsi :

- 1) Le vendeur propose aux acheteurs un produit à vendre à un certain prix initial
- 2) Chaque acheteur augmente le prix du produit avec une mise et le transmet au vendeur. Chaque acheteur possède sa propre mise qui est différente de celles des autres, et son propre prix maximum. Si le prix du produit dépasse le prix maximum d'un acheteur, ce dernier s'arrête.
- 3) A chaque étape, le vendeur, après avoir reçu tous les prix proposés, envoie aux acheteurs le plus grand prix reçu.
- 4) Les étapes 2 et 3 sont répétées jusqu'à ce que tous les acheteurs s'arrêtent, ou bien le temps de l'enchère se termine.
- 5) A la fin du processus d'enchère, si le dernier prix reçu par le vendeur est plus haut qu'un prix de reserve (connu seulement par le vendeur), le produit sera vendu au gagnant.

Interface graphique de l'application :

L'interface comprend trois parties : Vendeur, acheteur et resultat.

Dans la partie « vendeur », l'utilisateur peut sélectionner un produit parmi ceux proposés, introduire son prix initial, son prix reserve et le nombre d'acheteurs.

Dans la partie « acheteur » il introduit le prix maximum et la mise de chaque acheteur et ajoute ce dernier.

Quand il lance l'enchère, les résultats du déroulement sont affichés, comme le montre la figure suivante :

The screenshot shows a Java Swing window titled "Application d'enchère". It is divided into three main sections:

- Partie Vendeur:** Contains four input fields: "Produit" (a dropdown menu with "Ecran" selected), "Prix" (text input with "1000"), "Prix reserve" (text input with "5000"), and "Nombre d'acheteurs" (a dropdown menu with "3" selected).
- Partie acheteur:** Contains two input fields: "Prix seuil" (text input with "4000") and "Mise" (text input with "250"). Below these is a button labeled "Ajouter acheteur".
- Resultat:** A large text area at the bottom displays the log of the auction process. Above it is a button labeled "Lancer l'enchère".

The log text in the "Resultat" area shows the following sequence of events:

```
je suis Acheteur3 j'ai reçu le produit Ecran avec le prix 1000.0 je propose le nouveau prix= 1250.0
je suis Acheteur1 j'ai reçu le produit Ecran avec le prix 1000.0 je propose le nouveau prix= 1100.0
je suis Acheteur2 j'ai reçu le produit Ecran avec le prix 1000.0 je propose le nouveau prix= 1200.0
-----Je suis Vendeur j'ai reçu un prix max =1250.0-----
je suis Acheteur3 j'ai reçu le produit Ecran avec le prix 1250.0 je propose le nouveau prix= 1500.0
je suis Acheteur2 j'ai reçu le produit Ecran avec le prix 1250.0 je propose le nouveau prix= 1450.0
je suis Acheteur1 j'ai reçu le produit Ecran avec le prix 1250.0 je propose le nouveau prix= 1350.0
-----Je suis Vendeur j'ai reçu un prix max =1500.0-----
je suis Acheteur1 j'ai reçu le produit Ecran avec le prix 1500.0 je propose le nouveau prix= 1600.0
```

Partie 3:

Par manque de temps, nous n'avons pas pu refaire la deuxième partie avec la plateforme JASON, mais nous nous sommes documentées sur les deux plateformes et nous avons pu tirer les caractéristiques les plus importantes de chacune en se basant sur des références et documents portant sur des problèmes où elles ont été invoquées.

JASON :

Un agent de Jason a la capacité d'effectuer des cycles de raisonnement, c'est-à-dire l'architecture de l'agent traite un cycle de sens-plan-acte, qui permet aux agents d'évaluer les plans qui sont déclenchés pour exécution chaque fois qu'un événement se produit. C'est différent par rapport à JADE, qui fournit plutôt l'abstraction d'agent comme un ensemble de comportements avec des capacités de communication. Pour cette raison, il est non requis pour les agents de Jason de prévoir un traitement spécifique d'événements sociaux; ils peuvent être modélisés comme des événements de Jason normaux.[6]

Concepts fondamentaux :

JADE :

- ✓ Chaque instance de JADE est appelée conteneur , et peut contenir plusieurs agents.
- ✓ Chaque agent est constitué de plusieurs classes.
- ✓ Un conteneur par machine.
- ✓ Un ensemble de conteneurs constituent une plateforme.
- ✓ Chaque plateforme doit contenir un conteneur spécial appelé *main-container* et tous les autres conteneurs s'enregistrent auprès de celui-là dès leur lancement.[7]
- ✓ 1 agent = 1 *thread*
- ✓ Mobilité et clonage des agents - Les agents peuvent migrer à travers des conteneurs
- ✓ Les agents ont des comportements - Tâches exécutées simultanément
- ✓ Planification du comportement
 - Non préemptive, mais "coopérative"
 - Les comportements sont exécutés de manière "séquentiellement parallèle" [9]

JASON :

- ✓ Développement d'agents cognitifs avec l'architecture BDI (Beliefs-DesiresIntentions)
 - Les croyances sont des informations dont dispose l'agent sur le monde
 - Les désirs sont tous les états de choses possibles que l'agent peut aimer accomplir
 - Les intentions sont les états de faits sur lesquels l'agent a décidé de travailler.
- ✓ Raisonnement pratique (orienté vers les actions): délibération et raisonnement moyens-fins
 - Croyances, objectifs, plans
 - AgentSpeak: un langage de programmation pour agents BDI guidé par raisonnement pratique
 - Basé sur la programmation logique [9]

Avantages :

JADE :

Son architecture de communication tente d'offrir une messagerie flexible et efficace, en choisissant de manière transparente le meilleur transport disponible et en exploitant la technologie d'objet distribué intégrée dans l'environnement d'exécution Java. JADE utilise un modèle d'agent et une implémentation Java qui permettent une bonne efficacité d'exécution, la réutilisation de logiciel, la mobilité d'agent et la réalisation de différentes architectures d'agent. [8]

JASON :

Jason permet de mettre en œuvre des systèmes multi-agents complexes grâce à son extensibilité. La combinaison d'architectures et d'environnements d'agents personnalisés permet à l'utilisateur de simuler des scénarios réalistes à l'aide d'agents logiciels. [10]

Aussi, il possède une sémantique claire et bénéficie d'un grand nombre d'outils d'analyse, et comme il s'agit d'un langage de programmation polyvalent, il peut fonctionner avec des structures de données personnalisées et offrir une plus grande flexibilité en ce qui concerne la représentation des données. [11]

Désavantages :**JASON :**

L'utilisation d'un langage de programmation logique rend Jason potentiellement moins intuitif, car les langages de programmation impératifs sont plus populaires. [10]

Aussi, certaines situations réelles pouvant être rencontrées lors de l'utilisation de systèmes multi-agents dans des environnements ouverts peuvent être difficiles à exprimer dans une représentation basée sur la logique. Un autre inconvénient est que le traitement lors des structures de données personnalisées doit également être effectué de manière personnalisée par le programmeur. [11]

Conclusion :

Ce projet nous a permis de pratiquer les connaissances théoriques acquises dans le cours du module et voir une façon d'adopter la technologie des systèmes multi-agent dans des situations\problèmes réels, et ceci en :

- Implémentant un système expert à base de règles, dédié au domaine médical, sous JAVA.
- Implémentant un processus d'enchère entre agents intelligents à l'aide de la plateforme JADE dédiée aux systèmes multi-agent, et ceci en développant une application sous JAVA.
- Etablissant une comparaison entre les plateformes JADE et JASON.

REFERENCES :

- [1] «*Constructing Intelligents with java second edition*» Joseph P.Bigus, Jennifer Bigus
- [2] Webopedia «*Rule-based system*»: [<https://www.webopedia.com>]
- [3] «*Developing Multi-Agent Systems with Jade*» Fabio Bellifemine, Giovanni Caire, Dominic Greenwood
- [4] «*Multi-agent systems and decentralized artificial superintelligence* »
[<https://arxiv.org/ftp/arxiv/papers/1702/1702.08529.pdf>]
- [5] «*Implementation of Distributed Multi Agent System using JADE Platform* »
[<https://pdfs.semanticscholar.org/e015/c0ac1b1fc51db6040ea43c24b1990cd8dd9e.pdf>]
- [6] «*Programming JADE and Jason agents based on social relationships using a uniform approach* » [http://www.di.unito.it/~argo/papers/2015_MFSC.pdf]
- [7] «*JADE:Java Agent Development Framewrok* » Présenté par : Ameer Mohamed El Amine -USTHB
- [8] «*Developing Multi-agent Systems with JADE* » [<https://www.researchgate.net/>]
- [9]«*Multi-Agent Systems Software Platforms/Frameworks*»
[https://paginas.fe.up.pt/~eol/AIAD/aulas/MAS_sw.pdf]
- [10] «*Multi-Agent Systems: An Investigation of the Advantages of Making Organizations Explicit*» - Andreas Schmidt Jensen
[<https://pdfs.semanticscholar.org/a816/c1f5c1d80879ec72a5a99fc798f313566f94.pdf>]
- [11] «*A Comparison Between Jason and F-Sharp Programming Languages for the Enactment of Business Agents* »

[https://www.researchgate.net/publication/306541644_A_Comparison_Between_Jason_and_F-Sharp_Programming_Languages_for_the_Enactment_of_Business_Agents]