

<u>Université 20 Août 1955 – Skikda</u>		<u>جامعة 20 أوت 1955 - سكيكدة</u>
<u>Faculté des Sciences</u>		<u>كلية العلوم</u>
<u>Departement d'informatique</u>		<u>قسم الإعلام الآلي</u>

**Mémoire De fin d'étude en vue de l'obtention du  
Diplôme de Master en Informatique  
Option : Génie Logiciel Avancés et Application**

**Sujet :**

**Application pour le maintien à domicile des  
personnes âgées**

**Réalisé par les étudiants :**

- Lekkam Halima
- Tahir Bouchra

**Dirigé par :**

Kissom Yacine

**Année Universitaire 2021-2022**



# Remerciement



En premier lieu nos plus sincères remerciements vont au **bon Dieu** tout puissant qui nous avoir donné la grande volonté et un savoir adéquat pour mener à bien ce modeste travail.

Nos remerciements sont adressés également à nos **chers parents** pour tous les sacrifices consentis à notre égard et leur énorme soutien.

Nos vifs remerciements, s'adressent à notre Encadreur **DE *KISSOUM YACINE*** Dont les conseils et orientation nous ont été précieusement utiles pour la réalisation de ce projet.

Nous remercions très respectueusement tous les membres de jury d'avoir accepté de juger notre travail et de l'avoir enrichi.

Enfin, nos remerciements s'adressent à toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail.

« MERCI »

# Dédicaces

## *Je dédie ce projet :*

*A ma chère mère, Saliha / A mon cher père, Mechatí  
Qui n'ont jamais cessé, de formuler des prières à mon égard,  
de me soutenir et que je puisse atteindre mes objectifs.  
A mon frère, Boubakra El Sadik / A mes chères sœurs, Fatima et  
leur fille Baraa Chaïne, Adjar, Zinebe, Nouora, Nardjes.*

*A ma chère grand-mère, Yakouta*

*A Qui je souhaite une bonne santé.*

*A mon cher oncle, ZEGGARI Mechatí*

*Pour son entente et sa sympathie.*

*Pour leurs indéfectibles soutiens et leur patience infinie.*

*A mes chères cousines, Sabrina, Yamina, Nihed, Mona,*

*Amani, Bouchra, Zolikha.*

*Qui m'a aidé et supports dans les moments difficiles.*

*A mes chères amies, Sameh, Assila, Rayene, Nesrine,*

*Bouchra, Adjar, Aymen.*

*Pour leurs aides et supports dans les moments difficiles. A  
toute ma famille, Lekkam, Zegarri, Taoutaou tous mes  
autres ami(e), A tous ceux que j'aime et ceux qui m'aiment.*

*Halima*

# Dédicaces

## *Je dédie ce projet :*

*A l'âme de ma mère « Hayet » qui nous a quitté cette année*

*Aujourd'hui je suis Réalise le rêve que tu Souhaitais depuis des années, même si tu n'étais pas avec moi, je sais que tu es fière de moi.*

*A mon cher père, MOURAD*

*Qui n'a jamais cessé, de formuler des prières à mon égard, de me soutenir et que je puisse atteindre mes objectifs.*

*A mes frère, islem et iyed/A mes chères sœurs, sara et rouya. Pour leurs aides et supports dans les moments difficiles.*

*A mon cher oncle, Sayoud hacene.*

*A mon cher fiancé, Marouane.*

*A mes chères amies, Sameh et halima.*

*A mes chères collègues de travail.*

*Qui m'a aidé et supports dans les moments difficiles.*

*A toute ma famille, Sayoud et Tahir*

*A tous mes autres ami(e), A tous ceux que j'aime et ceux qui m'aiment.*

*Bochra.*

# ملخص

في حياتنا اليومية ويجلب العديد من التطبيقات المبتكرة ، والتي (AI) يتم دمج الذكاء الاصطناعي تجعل من الممكن تحسين مجتمعنا من حيث المساعدة والسلامة والراحة. يمنح هذا الطموح تقنيات المعلومات والاتصالات مكانة جديدة، وهي أن تكون في خدمة الإنسانية ، تتجاوز بكثير الدور الاختزالي لأداة الحساب التي غالبًا ما تُنسب إليهم

الذكاء المحيط هو نتاج الحوسبة التي ، من خلال دفع الحدود التكنولوجية إلى الوراء بطريقة تخريرية ، تثير التساؤل حول مفهوم نظام المعلومات أو الكمبيوتر: من نشاط معالجة يركز حصريًا على المستخدم إلى نهاية القرن العشرين ، يهدف الذكاء المحيط إلى التحكم في التفاعلات بين الكائنات المتصلة والبشر.

الهدف من مشروعنا هو تطوير نظام دعم منزلي للمسنين. يتعلق هذا النشاط بالمساعدة في الأعمال اليومية للحياة ، أو المساعدة في الاندماج الاجتماعي ، لكبار السن

في هذه المذكرة، أظهرنا الوصف العام لتطبيقنا وشرحنا مرحلة تصميم هذا المشروع بهذه SMA. هي منهجية تغطي إلى حد كبير عملية تطوير PASSI. PASSI المخططات المختلفة لطريقة من تعريف الأهداف إلى التصميم والتنفيذ النهائي للتطبيق SMA يجعل من الممكن متابعة تطور بفضل تحديد الأدوار الدقيقة لكل SMA علاوة على ذلك، تضيف هذه المنهجية بعدًا تنظيميًا على وكيل في النظام ، وهي خاصية مهمة في حالتنا. أخيرًا ، نظرًا للعديد من المدخلات / المخرجات التي يمكن أن تحدث في النظام الذي نقوم بتصميمه ، كان علينا اختيار منهجية تأخذ هذا الجانب الديناميكي في الاعتبار.

من المتوقع أن يخفف هذا المشروع موظفي المستشفى من المهمة التي تستغرق وقتًا طويلاً المتمثلة في مراقبة البيانات باستمرار وتزويد المرضى برعاية صحية في كل مكان وشخصية.

الكلمة الرئيسية: الذكاء الاصطناعي المحيط بالذكاء الاصطناعي ، الذكاء الاصطناعي ، PASSI ، أنظمة متعددة الوكلاء : SMA .

# Résumé

L'Intelligence artificielle (IA) s'intègre à notre quotidien et apporte de nombreuses applications innovantes, qui permettent d'améliorer notre société en matière d'assistance, de sécurité et de confort, Cette ambition confère aux Technologies de l'information et de la Communication un nouveau statut, celui d'être au service de l'Humanité, bien au-delà du rôle réducteur d'outil de calcul qui leur est souvent attribué.

L'intelligence ambiante est le produit de l'informatique qui en repoussant les limites technologiques de manière disruptive, remet en cause le concept même de système d'information ou d'ordinateur : d'une activité de traitement exclusivement centrée sur l'utilisateur jusqu'à la fin du XXe siècle, l'intelligence ambiante vise à régir les interactions entre objets communicants et humains.

L'objectif de notre projet est de développer un système de maintien, a domicile des personnes âgées. Cette activité concerne l'assistance dans les actes quotidiens de la vie, ou l'aide à l'insertion sociale, aux personnes âgées.

Dans ce mémoire, nous avons montré la description générale de notre application et expliqué la phase de conception de ce projet avec ces différents diagrammes de la méthode PASSI. PASSI est une méthodologie qui couvre en grande partie le processus de développement d'un SMA. Elle permet de suivre l'évolution du SMA depuis la définition des objectifs jusqu'à la conception et l'implémentation final de l'application.

De plus, Cette méthodologie apporte une dimension organisationnelle au SMA grâce à la définition de rôles précis pour chaque agent du système, une caractéristique importante dans notre cas. Enfin, au vu des nombreuses entrées/sorties pouvant avoir lieu dans le système que nous concevons, nous nous devons de choisir une méthodologie qui prend en compte cet aspect dynamique.

Ce projet devrait permettre de soulager le personnel hospitalier de la tâche de surveillance constante des données, qui prend beaucoup de temps, et de fournir aux patients des soins de santé omniprésents et personnalisés.

Mot-clé : IA Intelligence Ambiante, L'Intelligence artificielle, PASSI, Systèmes multi-agents SMA.

# Abstract

Artificial Intelligence (AI) is integrated into our daily lives and brings many innovative applications, which make it possible to improve our society in terms of assistance, safety and comfort.

This ambition gives Information Technologies and Communication a new status, that of being at the service of Humanity, well beyond the reductive role of a calculation tool that is often attributed to them.

Ambient intelligence is the product of computing which, by pushing back the technological limits in a disruptive way, calls into question the very concept of the information system or computer: from a processing activity exclusively centered on the user to at the end of the 20th century, ambient intelligence aimed to govern the interactions between communicating objects and humans.

The objective of our project is to develop a home support system for the elderly. This activity concerns assistance in the daily acts of life, or assistance with social integration, for the elderly. In this thesis, we showed the general description of our application and explained the design phase of this project with these different diagrams of the PASSI method.

PASSI is a methodology that largely covers the process of developing an MAS. It makes it possible to follow the evolution of the MAS from the definition of the objectives to the design and the final implementation of the application. Moreover, this methodology brings an organizational dimension to the MAS thanks to the definition of precise roles for each agent of the system, an important characteristic in our case.

Finally, given the many inputs/outputs that can take place in the system we are designing, we had to choose a methodology that takes this dynamic aspect into account. This project is expected to relieve hospital staff from the time-consuming task of constantly monitoring data and provide patients with ubiquitous and personalized healthcare.

Keyword: AI Ambient Intelligence, Artificial Intelligence, PASSI, SMA multi-agent systems.



# Table des matières

Introduction générale

## **Chapitre 1 : l'intelligence ambiante**

Introduction .....	1
Définition.....	1
Architecture logicielle d'un système ambiant.....	2
L'intelligence artificielle dans les systèmes ambiants.....	2
Problématique de l'intelligence.....	3
Les objets de l'intelligence ambiante pour l'assistance à domicile.....	4
Les composants élémentaires de l'intelligence ambiante.....	5
L'intelligence ambiante maintenant accessible à tous.....	5
Conclusion.....	6

## **Chapitre 2 : La méthode PASSI**

Introduction.....	7
Définition.....	7
Conception avec la méthode PASSI.....	8
Spécification des besoins.....	10
La définition de la société d'agents.....	14
L'implémentation des agents.....	20
Le codage.....	22
Le deployment.....	24
Conclusion.....	25

### **Chapitre 3 : Analyse et conception**

Introduction.....	26
Conception avec la méthode PASSI.....	26
Spécification de besoin.....	26
Tache 1.....	27
Tache 2.....	28
Tache 3.....	30
Tache 4.....	32
Conclusion.....	32

### **Chapitre 4 : Implémentation**

Introduction.....	33
Présentation du langage Java.....	33
Les caractéristiques de java.....	33
Présentation.....	34
Définition 1.....	34
Définition 2.....	34
Caractéristiques.....	35
Plateforme JADE.....	35
Conteneurs.....	36
Outils.....	36
Architecture de la plateforme Jade.....	37
L'interface graphique de JADE.....	38
Installation et configuration.....	39
Lancement de la plateforme.....	39
Lancement d'agent.....	39
Captures d'écran sur quelques interfaces de l'application.....	39
Conclusion.....	42

# Table des figures

<b>Figure 1</b> : Architecture d'un système Ami.....	3
<b>Figure 2</b> : Les cinq modèles/phases de la méthode PASSI.....	9
<b>Figure 3</b> : Le processus PASSI complet.....	9
<b>Figure 4</b> : Le diagramme de description de domaine.....	10
<b>Figure 5</b> : La phase des exigences système « La <a href="#">description du domaine</a> ».....	11
<b>Figure 6</b> : Le diagramme d'identification de l'agent .....	11
<b>Figure 7</b> : La phase des exigences système « <b>L'identification des agents</b> ».....	12
<b>Figure 8</b> : Le diagramme d'identification des rôles.....	12
<b>Figure 9</b> : La phase des exigences système « <a href="#">L'identification des rôles</a> ».....	13
<b>Figure 10</b> : Le diagramme de spécification des taches.....	14
<b>Figure 11</b> : Le diagramme de description d'ontologie de domaine.....	15
<b>Figure 12</b> : Diagramme de description de l'ontologie de communication.....	17
<b>Figure 13</b> : Diagramme de description des rôles.....	19

<b>Figure 14</b> : Diagramme de séquence.....	20
<b>Figure 15</b> : Le diagramme MASD.....	21
<b>Figure 16</b> : Le diagramme MABD.....	22
<b>Figure 17</b> : La phase de code.....	23
<b>Figure 18</b> : Achèvement du code Fragment de référence-Aspect procédural.....	24
<b>Figure 19</b> : The Déploiement Configuration diagramme.....	25
<b>Figure 20</b> : diagramme de description du domaine.....	28
<b>Figure 21</b> : diagramme d'identification des agents.....	29
<b>Figure 22</b> : diagramme d'identification des rôles pour « Login ».....	30
<b>Figure 23</b> : diagramme d'identification des rôles pour « système ».....	31
<b>Figure 24</b> : diagramme de spécification des tâches.....	32
<b>Figure 25</b> : La plate-forme JADE.....	36
<b>Figure 26</b> : Architecture logiciel de la plate-forme JADE.....	38
<b>Figure 27</b> : Interface graphique de JADE.....	38
<b>Figure 28</b> : Interface département (société).....	39
<b>Figure 29</b> : Interface design preview (Login).....	40

<b>Figure 30</b> : Interface authentification (Docteur/Patient).....	40
<b>Figure 31</b> : Interface accueil pour patient.....	41
<b>Figure 32</b> : Interface accueil médecin .....	41
<b>Figure 33</b> : Interface recherche / ajout des patients .....	41
<b>Figure 34</b> : Interface d'enregistrement des mesures.....	42

# Introduction générale

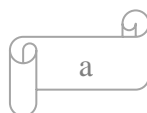
L'intelligence ambiante concerne l'utilisation de technologies pour calculer, détecter, afficher, communiquer et interagir afin de fournir des services dans des environnements humains. Différentes facettes de ce problème ont été abordées sous une variété de noms, y compris l'informatique omniprésente, l'informatique omniprésente, l'informatique en voie de disparition et l'internet des objets. Quel que soit son nom, le domaine se définit par son objectif principal : fournir des services et des dispositifs capables de s'adapter aux besoins des individus et au contexte social. Cela comprend diverses applications telles que l'aides aux personnes à adopter des modes de vie plus économes en énergie ,l'amélioration de la qualité de vie des personnes handicapées ,l'aide aux personnes âgées pour rester indépendantes et l'aide aux familles avec des services de sécurité, de divertissement et avec des outils de gestion du cout de la vie.

Notre population est de plus en plus vieillissante .Plus que par le passé donc, le soutien aux seniors est devenu une préoccupation majeure. Mais les informations relayées dans les journaux montrent que leurs besoins ne sont pas toujours pris en compte .Des cas de violence physique, d'abandon ou d'escroquerie sur les personnes âgées sont devenus courants. Mais vous, que faites-vous personnellement pour combler les besoins des personnes âgées ? A travers ce dossier, je propose quelques activités et impressions qui ravirons sans doute vos propres parents ..... Et vous-même.

L'objectif de notre projet est développer un système d'assistance aux personnes âgées malades vivants seules à domicile.

Cette activité concerne l'assistance dans les actes quotidiens de la vie , ou l'aide à l'insertion sociale , aux personnes âgées, handicapées ou atteintes de pathologies chroniques, ayant besoin de telles prestations à leur domicile .Elle Recouvre également l'aide personnelle à domicile aux familles fragilisées, Cela va permettre aux patients de contrôler leur santé chaque jour et faciliter l'interaction virtuelles patients/médecins à distance via leurs Smartphones ou tablettes . Ce système constitué d'une application de surveillance de santé d'une personne malade.

Ce mémoire est composé de quatre chapitres :



## **Chapitre 1 : L'intelligence ambiante**

Dans ce chapitre on parle sur l'intelligence ambiante dans notre vie.

## **Chapitre 2 : La méthode PASSI**

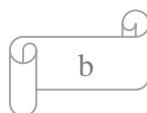
Dans ce chapitre montre l'ensemble de définition de la méthode PASSI avec des différents diagrammes.

## **Chapitre 3 : Analyse et conception**

Nous montre dans ce chapitre une description générale de notre application et expliquer la phase de conception de ce projet avec ces différents diagrammes avec le langage PASSI.

## **Chapitre 4 : Implémentation**

Nous présentons la réalisation de notre application en détail avec des captures d'écran.



# Chapitre1 : intelligence ambiante

## 1. Introduction :

L'intelligence n'est pas présente qu'humainement, elle est présente dans les nouvelles technologies que nous utilisons aujourd'hui comme par exemple les Smartphones, les smart homes les smart Tv ..... Tous ce qui est intelligent dans notre entourage, notre maison, dehors ou au travail c'est de l'intelligence ambiante. Cette intelligence est utilisée dans plusieurs domaine pour faciliter la communication humaine, améliorer la qualité de vie et plus particulièrement dans le domaine de la santé comme par exemple E-Heath.

Le terme intelligence ambiante (AMI) a été inventé par la Commission européenne en 2001 par l'un de ses groupes consultatifs des programmes (the Europe an Community's Information Society Technologie (USTAG)), puis le terme a été mis à jour en 2003. Mais bien que le terme Am né en Europe, son utilisation et ses objectifs ont été adoptées au cours des dernières années, partout dans le monde, avec de nombreux projets et programmes de recherche connexes. C'est un domaine de recherche très actif qui a fait l'objet et programmes de recherche d'articles et d'éditions spéciales de revues (par exemple, IEEE Intelligent Systems Magazine, Journal of Ambient Intelligence and Smart Enivrements (JAISE)) et de plusieurs conférences spécialisées. Des exemples sont : la Conférence européenne sur l'intelligence ambiante, la Conférence internationale sur les Robots Ubiquitaires et l'intelligence ambiante.

Dans ce chapitre en a choisir de parler sur L'intelligence ambiante.

## 2. Définition :

L'intelligence ambiante est le produit de l'informatique qui en repoussant les limites technologiques de manière disruptive, remet en cause le concept même de système d'information ou d'ordinateur : d'une activité de traitement exclusivement centrée sur l'utilisateur jusqu'à la fin du XXe siècle, l'intelligence ambiante vise à régir les interactions entre objets communicants et humains. [1]

Milieu ayant la faculté de percevoir, de raisonner, d'agir et d'interagir afin de fournir des services améliorant la qualité de vie des êtres vivants et notamment des personnes. [1]

L'intelligence ambiante s'insère dans une vision futuriste des télécommunications et D'informatisation de la vie. Notre environnement futur sera entouré par différents systèmes et applications qui seront portés par les technologies réseaux et l'informatique pour fournir une aide intelligente sans intrusion humaine.

### 3. Architecture logiciel d'un système ambiant :

L'informatique ambiante est le point de convergence d'un nombre important de domaines : l'informatique distribuée, l'informatique mobile, l'intelligence artificiel, les réseaux, les capteurs, l'interaction homme machine, etc. [2]

Un environnement dit d'intelligence ambiante doit :

- Être contrôlable par des gens ordinaires ;
- Améliorer la qualité et la productivité dans un environnement de travail ;
- Simplifier le choix du consommateur ;
- Faciliter les contacts humains ;
- Aider à créer et à améliorer les compétences pour le travail ;
- Inspirer la confiance et à améliorer les compétences pour le travail ;
- Simplifier le choix du consommateur ;
- Faciliter les contacts humains ;
- Aider à créer et à améliorer les compétences pour le travail ;
- Inspirer la confiance et respecter de vie privée.

Actuellement, pour se connecter au réseau, il faut trouver un câble, passer les systèmes de sécurité et d'indentification.

Les chercheurs dans le domaine de l'informatique ubiquitaire et de l'intelligence ambiante s'intéressent de plus en plus aux architectures logicielles et aux interfaces hommes machines.

### 4. L'intelligence artificielle dans les systèmes ambiants :

Les fondements de l'ami tel que décrites par Mark Weiser [3] sont :

- Un réseau unifié,
- Des dispositifs bon marché ayant une faible consommation énergétique,
- Des interfaces pratiques,
- Des architectures logicielles permettant d'implémenter les applications d'Ami.

Les dispositifs d'un système d'Ami sont caractérisés par :

**Leur intégration:** Un grand nombre de dispositifs sont intégrés dans l'environnement.

**Leur sensibilité au contexte :** Les dispositifs peuvent reconnaître l'utilisateur et son profil.

**Personnalisation:** Les dispositifs sont bien adaptés en fonction des besoins des utilisateurs.

**Adaptatifs:** les dispositifs peuvent changer en fonction des demandes des utilisateurs.

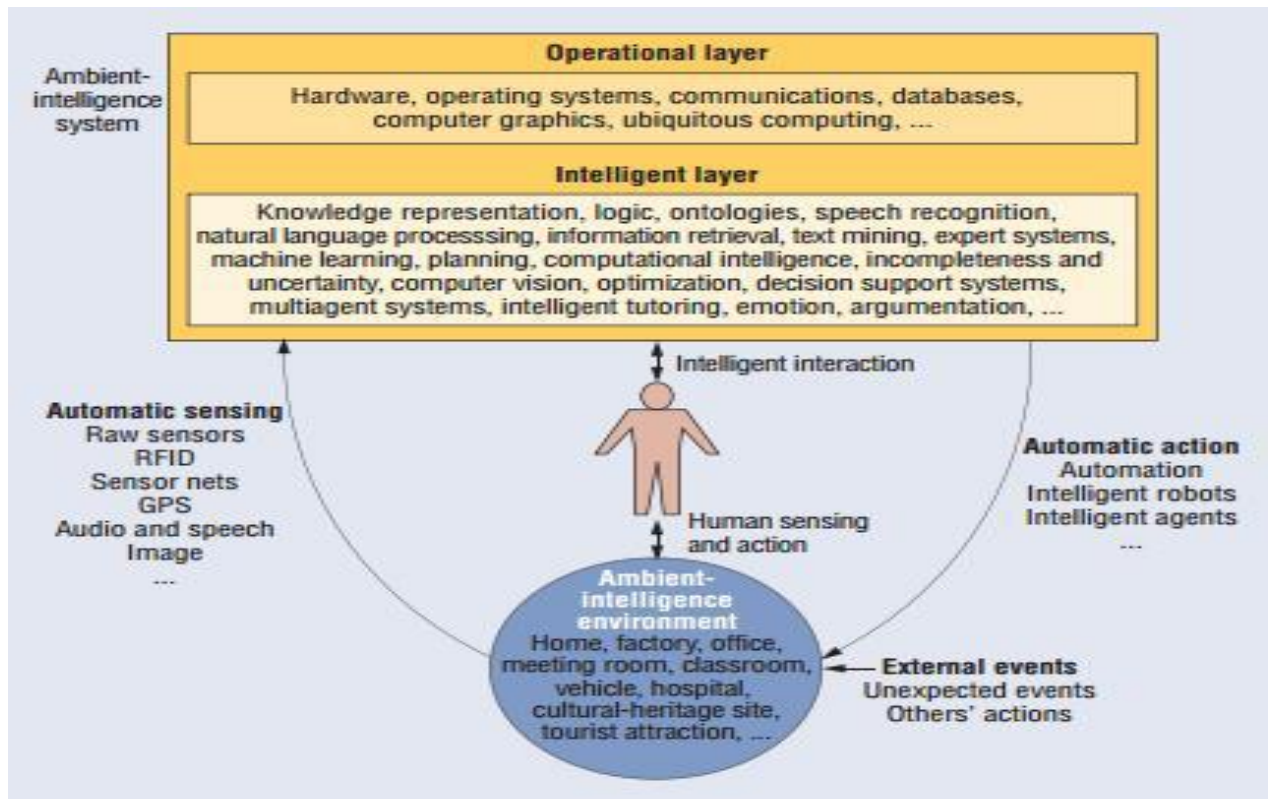


Figure 1: Architecture d'un système Ami

## 5. Problématique de l'intelligence ambiante :

Pour développer une application utilisant le paradigme d'Ami, nous devons d'abord répondre au 6Ws questions [3] :

### Qui (Who) :

L'identification d'un utilisateur et le rôle que l'utilisateur joue le système.

### Où (Where) :

Le suivi de l'endroit où un utilisateur ou un objet se situe géographiquement à chaque instant pendant le fonctionnement du système.

**Quand(When) :**

L'association des activités avec le temps est fondamentale pour construire une image réaliste de la dynamique d'un système.

**Quoi(What) :**

La reconnaissance des activités et des tâches que les utilisateurs effectuent est fondamentale afin de fournir une aide appropriée, si nécessaire. La multiplicité des scénarios possibles qui suivent une action rend la reconnaissance très difficile.

**Pourquoi(Why) :**

La capacité de déduire et de comprendre les intentions et les objectifs sous-jacents des activités est l'un des défis les plus difficiles, mais sans aucun doute une question fondamentale qui permet au système d'anticiper les besoins et servir les utilisateurs d'une manière sensible.

**Comment(How) :**

Les méthodes alternatives pour réaliser des choses dans l'environnement donné. Une architecture qui supporte les cinq précédents concepts seront à leur tour de faire avec l'appui des renseignements suffisants pour prendre les bonnes décisions en temps opportun.

Pour pouvoir trouver une réponse à ces 6Ws questions, il faut faire coopérer beaucoup de disciplines tels que l'interface homme-machine avancée, les réseaux, l'informatique ubiquitaire et l'intelligence artificielle. La réponse à ces questions ne peut se faire sans la mise en place de systèmes dotés de capacité de sensibilité au contexte.

**6. Les objets de l'intelligence ambiante pour l'assistance à domicile :**

L'intelligence ambiante utilise l'informatique diffuse des objets et l'intelligence artificielle pour faire émerger de nouvelles solutions technologiques qui facilitent le maintien à domicile des personnes âgées.

L'assistance à l'autonomie à domicile passe par un processus de transformation de l'habitat en habitat intelligent [4].

Le premier domaine d'application de l'intelligence ambiante sera l'assistance et la protection des personnes, en permettant de prévenir les accidents domestiques qui causent plus de 20 000 décès par an. Reliés à une IA, des capteurs « intelligents » peuvent détecter un danger imminent, comme un départ d'incendie domestique ou la présence d'un enfant près d'une fenêtre ouverte.

Dans notre société vieillissante, c'est une technologie précieuse, qui améliore le soin apporté aux personnes âgées dépendantes et augmente leur autonomie à domicile. Des systèmes d'analyse de vidéo peuvent ainsi reconnaître une chute ou une blessure et alerter les infirmiers ou les secours, afin qu'ils viennent en aide aux personnes en danger. [4]

## 7. Les composants élémentaires de l'intelligence ambiante :

L'intelligence ambiante met en œuvre trois éléments de base :

**L'ubiquité**: la capacité pour l'utilisateur d'interagir (activement ou passivement), n'importe où avec une multitude d'appareils interconnectés, de capteurs, d'activateurs, et plus globalement avec les systèmes électroniques embarqués (Embedded software) autour de lui. Tout cela à Travers des réseaux adaptés et une architecture informatique très distribuée.

**L'interaction naturelle**: l'accès aux services doit pouvoir se faire de la façon la plus naturelle/intuitive possible. A la différence de l'interface traditionnelle de l'univers informatique (dénommée WIMP, Windows, lacons, icones, menus et dispositif de pointage), l'interface personne-machine est multimodale. Elle s'articule autour de la reconnaissance vocale, de la reconnaissance et la manipulation d'objets réels.

**L'intelligence**: la faculté d'analyse du contexte et l'adaptation dynamique aux situations. Le système doit apprendre en se basant sur les comportements des utilisateurs afin de leur répondre au mieux. Cela implique capacités de stockage d'information, de traitement et algorithmes de modélisation et une approche dite d'intelligence artificielle. [5]

## 8. L'intelligence ambiante maintenant accessible à tous :

L'existence de ces caméras intelligentes est aujourd'hui possible grâce à trois facteurs principaux :

D'abord aux performances exceptionnelles atteintes par la reconnaissance visuelle, grâce à nouveautés apportées par le Deep-Learning.

Ensuite, le hardware, car pour créer et exploiter une application d'IA, des outils et des systèmes performants sont nécessaires. L'impressionnante augmentation de la puissance de calcul a permis la diminution de la taille de ce hardware, entraînant une baisse des prix et le rendant plus accessible.

Enfin, au moindre encombrement que représente le matériel on produit désormais des caméras miniatures et qui permet d'intégrer plus largement des applications d'IA aux caméras. L'intelligence devient ambiante car en devenant plus accessibles, les IAS vont pouvoir se répandre largement.

Le développement de l'intelligence ambiante représente une amélioration sans précédent pour notre société. Les entreprises qui développeront en premier cette technologie sont celles qui gagneront les appels d'offres d'un marché en pleine expansion, et contribueront à améliorer la qualité de vie de la société tout entière. [6]

## 9. Conclusion

La finalité de l'intelligence ambiante est l'amélioration, voire l'augmentation du monde réel pour offrir un tout adapté en toute circonstance à l'Homme. Autrement dit, il s'agit de créer des services et des dispositifs intelligents capables de répondre à des besoins individuels, collectifs et sociétaux.

Contrairement à l'informatique conventionnelle, il ne s'agit plus seulement de développer des artefacts technologiques qui améliorent l'efficacité de la personne sur son lieu de travail, mais d'améliorer le bien-être de l'individu et de la société. Cette ambition confère aux Technologies de l'information et de la Communication un nouveau statut, celui d'être au service de l'Humanité, bien au-delà du rôle réducteur d'outil de calcul qui leur est souvent attribué.

# Chapitre2: La Méthode PASSI

## 1. Introduction :

Depuis la fin des années 1990, l'Ingénierie Dirigées par les Modèles (IDM) permet de recentrer les processus de développement sur les modèles, et non plus sur le code. [7]

Les outils issus de ce domaine permettent de concevoir des langages de modélisation dédiés à un domaine d'expertise ; ils peuvent être associés à des éditeurs graphiques générés automatiquement et permettent d'obtenir des produits dérivés de ce modèle (comme du code) par transformation. [8]

Cette pratique nécessite une modélisation (un méta-modèle) des concepts du domaine source et une autre pour le domaine cible (par exemple, la plate-forme de développement). Cette approche a déclenché la définition de plusieurs méta-modèles dédiés aux domaines particuliers des SMA et de leur implémentation. On trouve, par exemple, AMAS-ML qui s'intègre dans ADELFE pour la définition d'agent coopératif. **Le méta-modèle CRIO [9], associé au processusASPECS, s'inspire de PASSI pour la définition de systèmes holoniques** et en reprend trois domaines sémantiques : le domaine pour la plateforme cible ; le domaine définissant le type particulier de SMA, holoniques pour CRIO ; et enfin, un domaine permettant de capturer les exigences que devra remplir le système et se focalisant sur la notion de rôle. Cette notion de rôle forme également le fondement du méta-modèle de GAIA, qui, en outre, met l'accent sur la notion de service que fournit un agent au sein d'une organisation ainsi que sur les responsabilités et activités liées à un rôle. Face à la diversité des approches et au nombre grandissant de méta-modèles engendrés, des essais de factorisation ont déjà été étudiés. [10]

Cependant, les possibilités offertes par l'IDM orientent actuellement les travaux plutôt vers une définition des méthodologies par parties composables et réutilisables [11].

Dans ce chapitre en a choisir de parler sur la méthode passsi de système multi agent

## 2. Définition

PASSI (Process for Agent Sociétés Spécification and Implémentation) est une méthode de conception d'application multi-agents. Elle intègre des modèles de développement et de conception basés sur les concepts de la programmation orientée objets et des concepts liés au paradigme agent. La méthode PASSI utilise les concepts des standards UML, XML et des notions dérivées des spécifications FIPA, ce qui la rend bien adaptée à FIPA-OS et JADE [12].

### 3. Conception avec la méthode Passi :

PASSI réutilise fortement UML. Par exemple, le modèle des besoins est exprimé grâce à des diagrammes de cas d'utilisation, de paquetages stéréotypés, de séquences (associés aux cas d'utilisation) et d'activité. Le modèle de société d'agents utilise des diagrammes de classes et de séquences. Les modèles d'implémentation d'agent et de codage utilisent des diagrammes de classes et d'activités classiques. Enfin, le modèle de déploiement utilise des diagrammes de déploiement UML. L'utilisation d'UML, comme pour ADELFE ou INGENIAS par exemple, implique le manque de formalisme (et donc de précision), et les connaissances ne peuvent être représentées qu'en utilisant des diagrammes de classes. [13]

PASSI est riche en ressources, et possède notamment un site expliquant le processus pas-à-pas ainsi que tous les modèles. Aucune expertise particulière n'est recommandée bien que la connaissance, non obligatoire, des spécifications FIPA soit préférable pour exploiter au mieux la méthode. PASSI n'est reliée à aucun langage particulier, bien que l'outil PTK (PASSI ToolKit) soit orienté vers Java afin d'être compatible avec JADE et FIPA-OS. Tous les diagrammes et stéréotypes spécifiques sont intégrés dans une extension au logiciel Rational Rose, afin de réaliser les modèles d'analyse et conception de manière ergonomique.

Le processus de développement est un processus pas à pas, mais reiterable. il est composé de cinq phases : les besoins, la définition de la société d'agents, modèle de mise en œuvre agent (l'implémentation des agents) modèle de codage, modèle de déploiement. Chaque phase est composée d'étapes, correspondant aux spécifications des cinq modèles présentés précédemment.

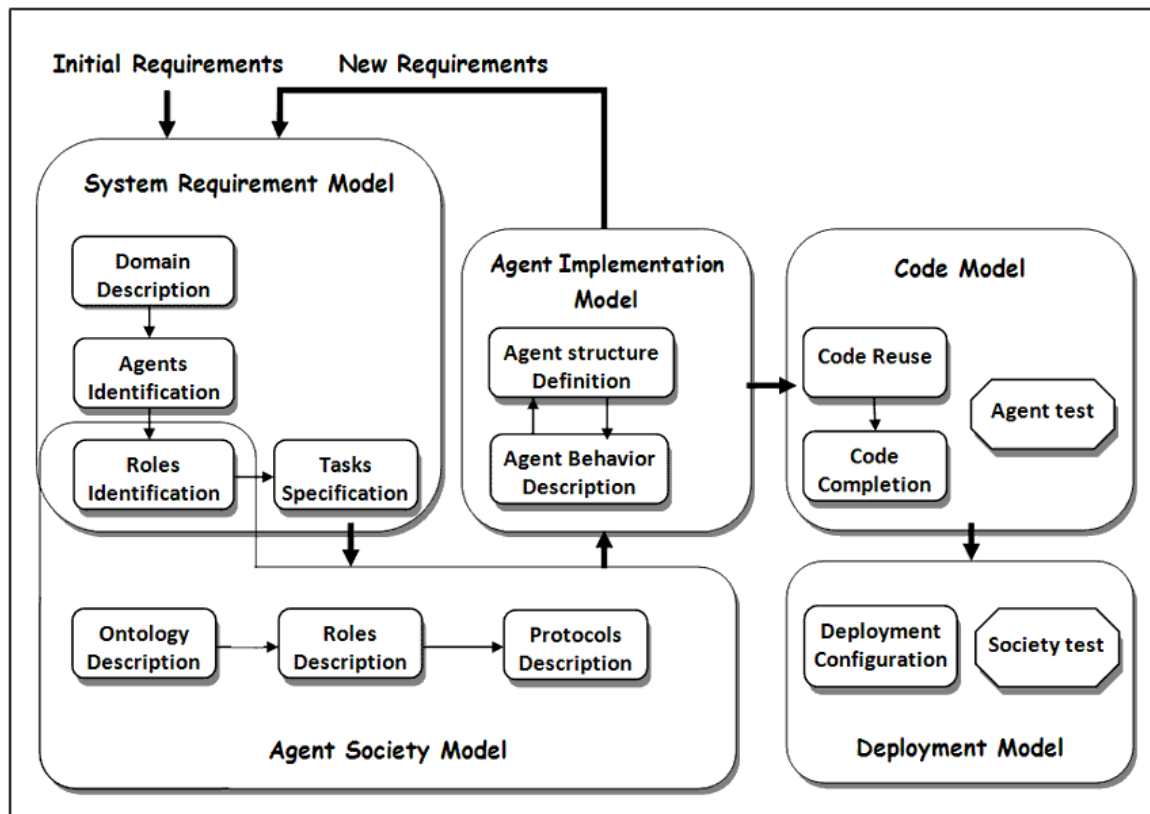


Figure 2: Les cinq modèles/phases de la méthode PASSI

Nous allons définir un fragment de méthode Domain Description, extrait de la méthodologie PASSI dont le processus est complètement représenté dans la figure suivante :

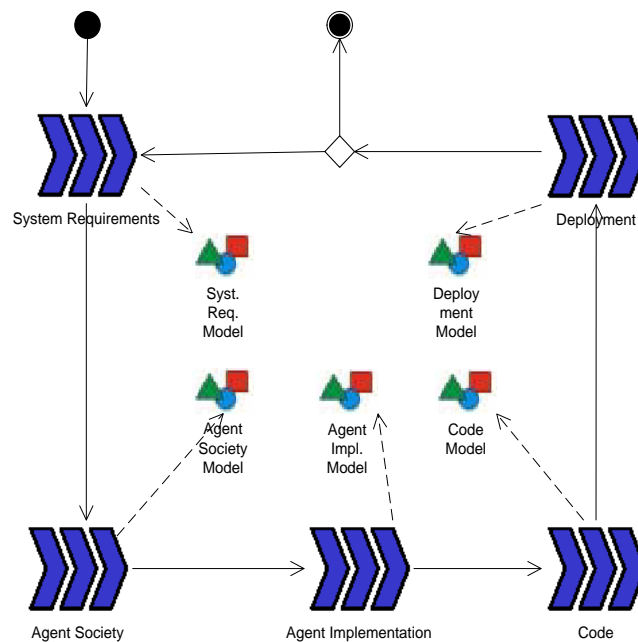


Figure 3 : Le processus PASSI complet

### 3.1. Spécification des besoins :

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

Après une étude détaillée de système, cette partie est réservée à la description des exigences des différents utilisateurs de l'application. Ces besoins se regroupent dans les modèles des besoins.

*3.1.1. La description du domaine : permet d'exprimer les besoins et de décrire le contexte du système grâce à une hiérarchie de cas d'utilisation ;*

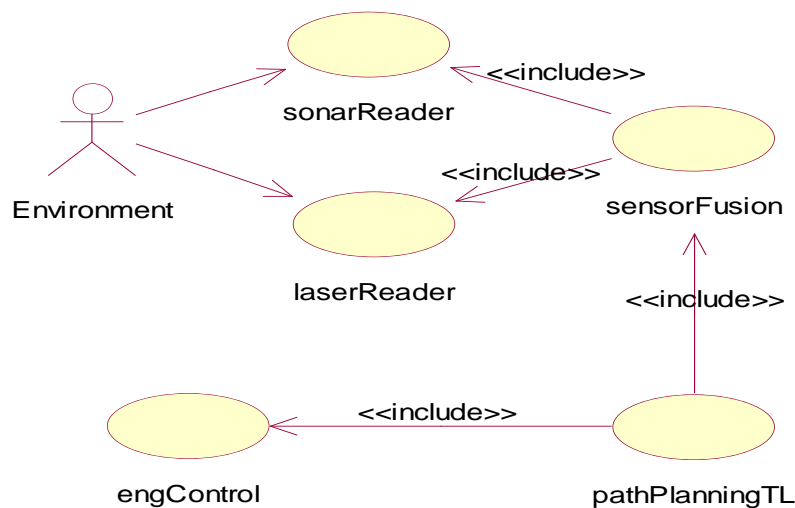


Figure 4 : Le diagramme de description de domaine

Nous allons définir un fragment de méthode que nous appelons "*La description du domaine*", extrait de la méthodologie PASSI dont le processus est complètement représenté dans la figure suivante ;

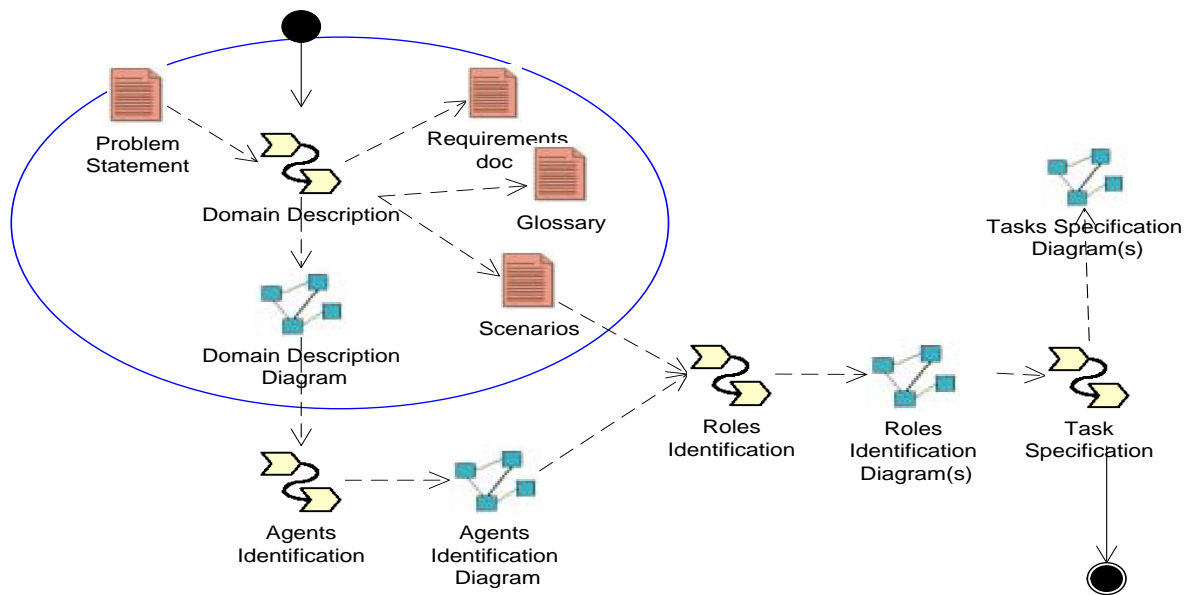


Figure 5: La phase des exigences système « La *description du domaine* ».

**3.1. 2.L'identification des agents :** est une étape peu présente dans les méthodes orientées agent, et est effectuée par regroupement en paquetages, qui correspondront aux agents ;

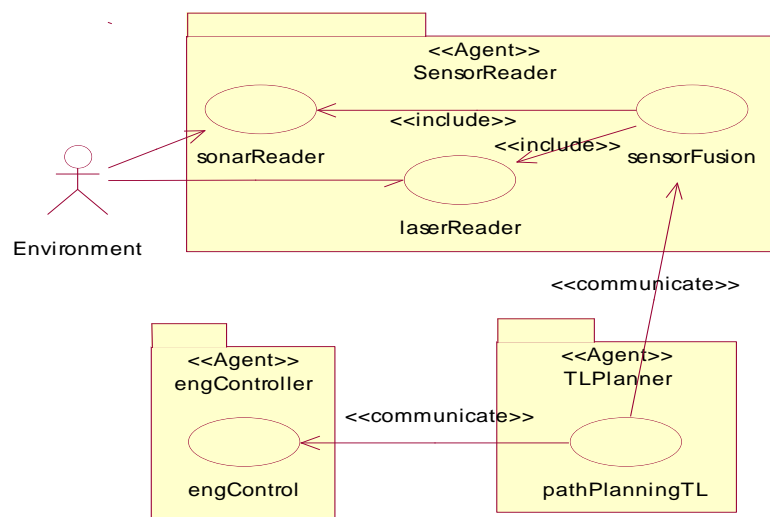


Figure 6: Le diagramme d'identification de l'agent

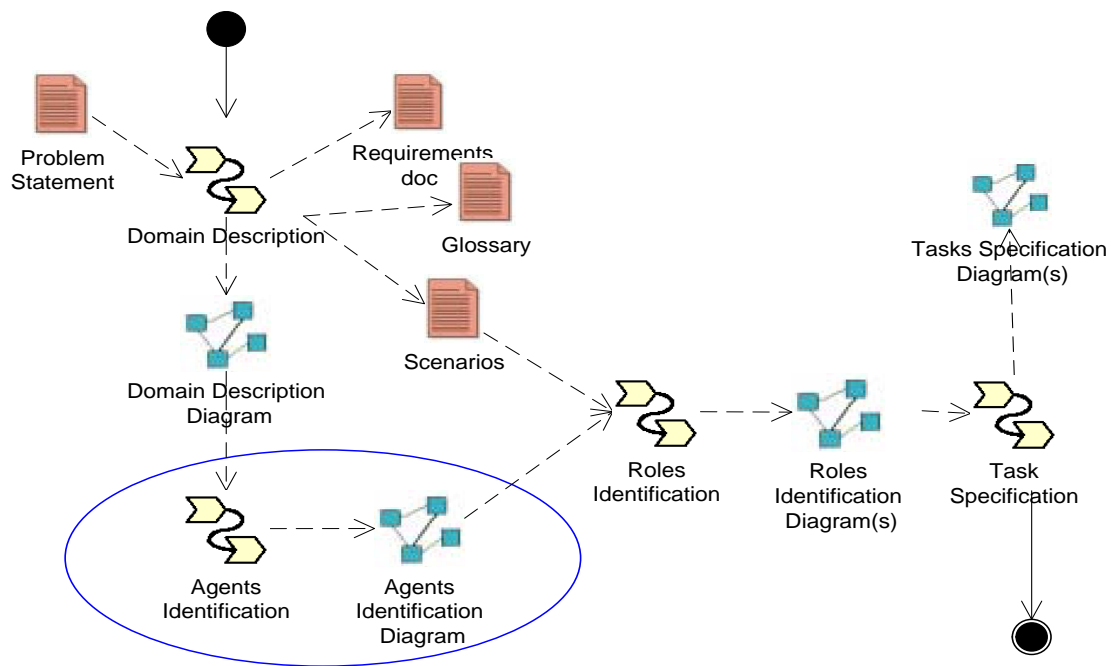


Figure 7 : La phase des exigences système « **L'identification des agents** ».

Nous allons définir un fragment de méthode que nous appelons "Agent Identification", extrait de la méthodologie PASSI dont le processus est complètement représenté dans la figure suivante ;

**3.1.3 .L'identification des rôles** : Les diagrammes de séquence décrivent toutes les interactions possibles entre les agents. Ces interactions décrivent un scénario de travail d'agent interagissant pour obtenir un comportement souhaité du système. Les rôles sont à la fois une description fonctionnelle et comportementale des agents, mais aussi une représentation des relations entre agents. Chaque agent peut appartenir à plusieurs scénarios, qui sont établis au moyen de diagrammes de séquence.

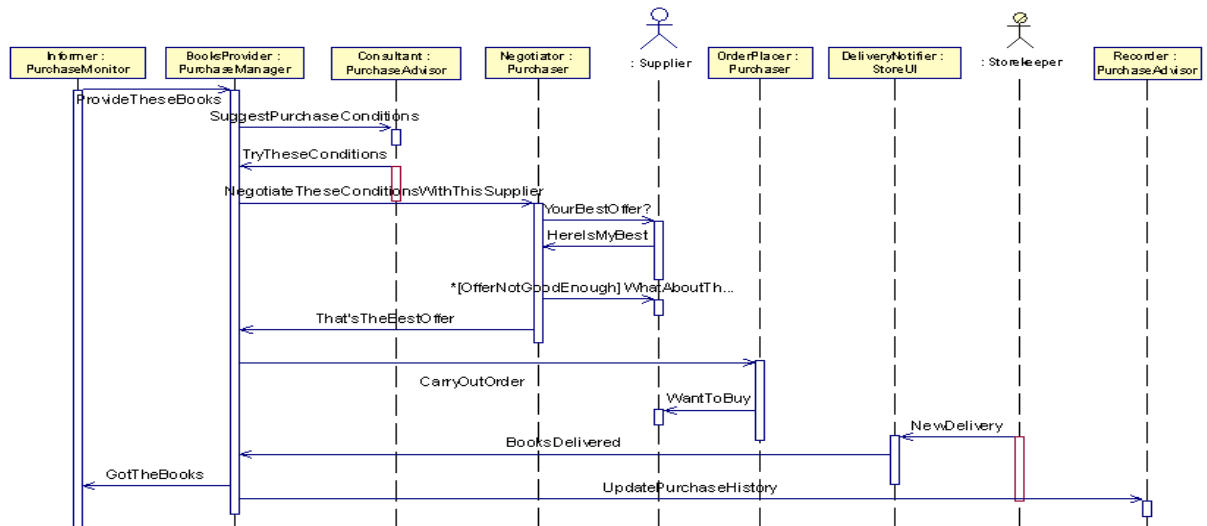


Figure 8: Le diagramme d'identification des rôles

Nous allons définir un fragment de méthode Identification des Rôles, extraite de la méthodologie PASSI dont le processus est entièrement représenté dans la figure suivante

Considérons la définition de travail « Identification des rôles » (l'ovale bleu) dont le but est de décrire tous les scénarios possibles d'agents en interaction travaillant pour obtenir un comportement requis du système.

Le modèle UML de cette partie du processus, le diagramme d'identification des rôles, est conçu selon une notation UML standard.

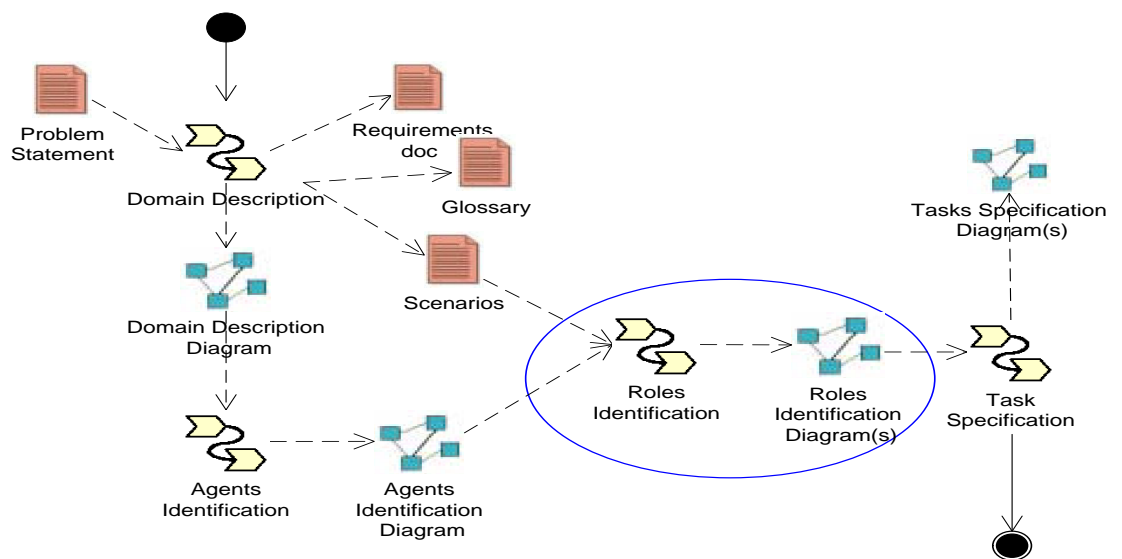


Figure 9 : La phase des exigences système « L'identification des rôles ».

**3.1.4. La spécification des tâches :** permet de définir les activités des agents en les reliant aux fonctionnalités fournies par le système ;

Un diagramme d'activité différent est dessiné pour chaque agent. Ce diagramme décrit comment l'agent peut utiliser ses tâches pour exécuter ses plans.

Chaque diagramme est composé de deux couloirs et contient des activités qui représentent généralement les tâches de l'agent. Le couloir de droite contient les tâches de l'agent que nous décrivons (Purchase Manager dans la figure ci-dessus), dans celui de gauche, nous pouvons trouver les tâches des autres agents qui interagissent avec celui-ci.

Les transitions dans le même couloir décrivent le flux de contrôle de différentes tâches tandis que les transitions d'un couloir à l'autre représentent les communications.

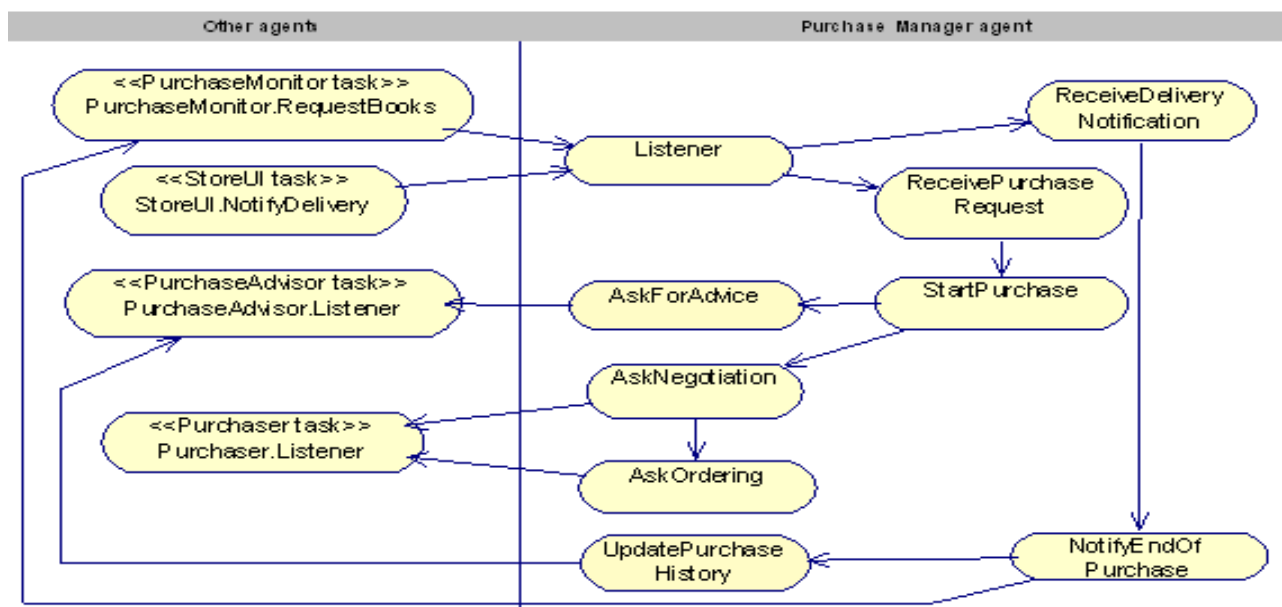


Figure 10 : Le diagramme de spécification des tâches

## 3.2. La définition de la société d'agents :

### 3.2. 1. Identification des rôles : (voir plus haut)

**3.2. 2. La description de l'ontologie :** Permet de décrire la société d'agents d'un point de vue ontologique suivant deux axes : ontologie du domaine (diagramme de classes appelé diagramme d'ontologie du domaine) et l'échange d'information entre agents (diagramme de classes appelé diagramme d'ontologie de communication).

**3.2. 2.1 ontologie du domaine :** L'ontologie est décrite (à l'aide d'un diagramme de classes) en termes de concepts (couleur de remplissage : jaune), de prédicats (couleur de remplissage : bleu clair) et d'actions (couleur de remplissage : blanc).

Les éléments de l'ontologie peuvent être liés à l'aide de trois relations standard UML :

- Généralisation : elle permet la relation de généralisation/spécialisation entre deux entités qui est l'un des opérateurs fondamentaux pour construire une ontologie.
- Association : elle modélise l'existence d'une sorte de relation logique entre deux entités. Il est possible de préciser le rôle des entités impliquées afin de clarifier la structure.
- Agrégation : elle peut être utilisée pour construire des ensembles où des restrictions de valeur peuvent être explicitement spécifiées ; dans la norme RDF du W3C, trois types d'objets conteneurs sont énumérés : le sac (une liste non ordonnée de ressources), la séquence (une liste ordonnée de ressources) et l'alternative (une liste de valeurs alternatives d'une propriété). Nous choisissons de considérer un sac comme une agrégation sans restriction explicite, une séquence est qualifiée par l'attribut ordonné alors que l'alternative est identifiée par le seul attribut de la relation.

Dans la figure précédente, nous avons une petite partie d'une ontologie de vision robotique. Mono Image est une spécialisation du concept Im Data avec un horodatage (grabbing time). L'agrégation ordonnée de deux images mono donne la Stéréo Image. Nous définissons l'action Give Image afin de permettre à un robot de demander une image. L'image doit être fournie par l'Acteur et envoyée au ResultReceiver (les deux agents). Des prédicats sont également définis par rapport à certains concepts existants (Is Image, Is St Image).

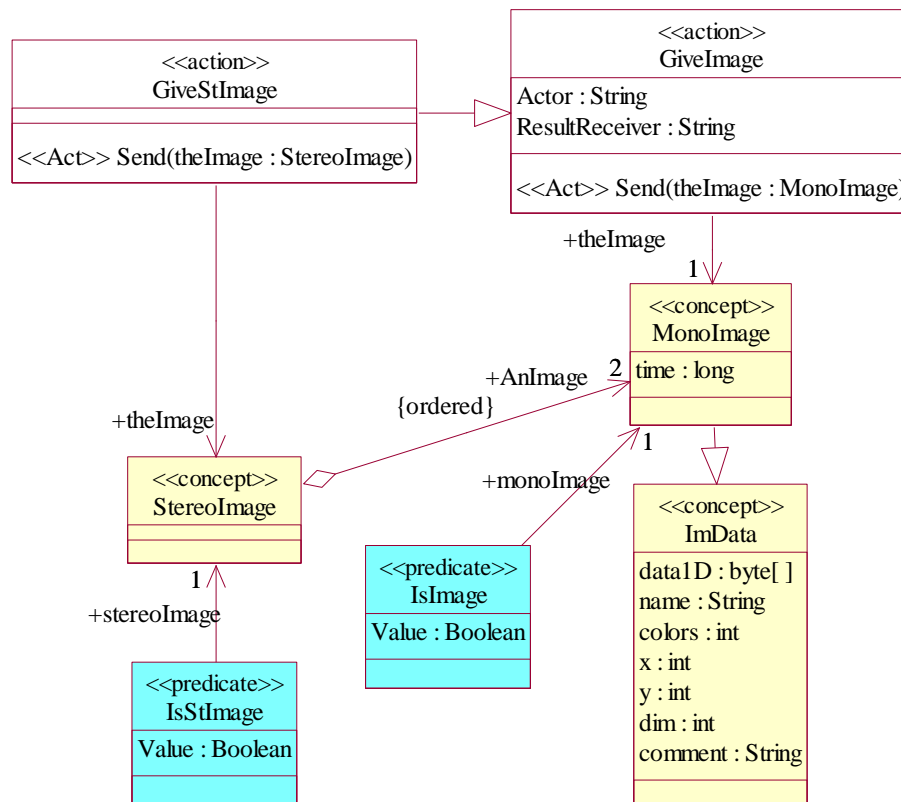


Figure 11: Le diagramme de description d'ontologie de domaine

### 3.2. 2.2 Description de l'ontologie de communication :

Le diagramme COD est un diagramme de classes et il est principalement composé de deux éléments : les agents et les communications.

Chaque agent (couleur de remplissage : jaune) est décrit en fonction de ses connaissances (morceaux de l'ontologie décrite dans le schéma précédent). Il existe une relation entre deux agents pour chaque communication dans laquelle ils sont impliqués.

Dans chaque relation, les rôles joués par les agents au cours de la communication sont également rapportés.

Chaque communication (couleur de remplissage : blanc) est représentée par la relation entre les deux agents et elle est détaillée dans la classe d'attributs de relation. La classe est identifiée par un nom unique (également rapporté dans la relation entre les deux agents) et elle est décrite par les champs ontologie, langage et protocole.

Le champ ontologie fait référence à un élément de la DOD (Domain Ontologie Description) ; les adresses de langue pour la langue du contenu de la communication tandis que le protocole indique le protocole d'interaction FIPA adopté.

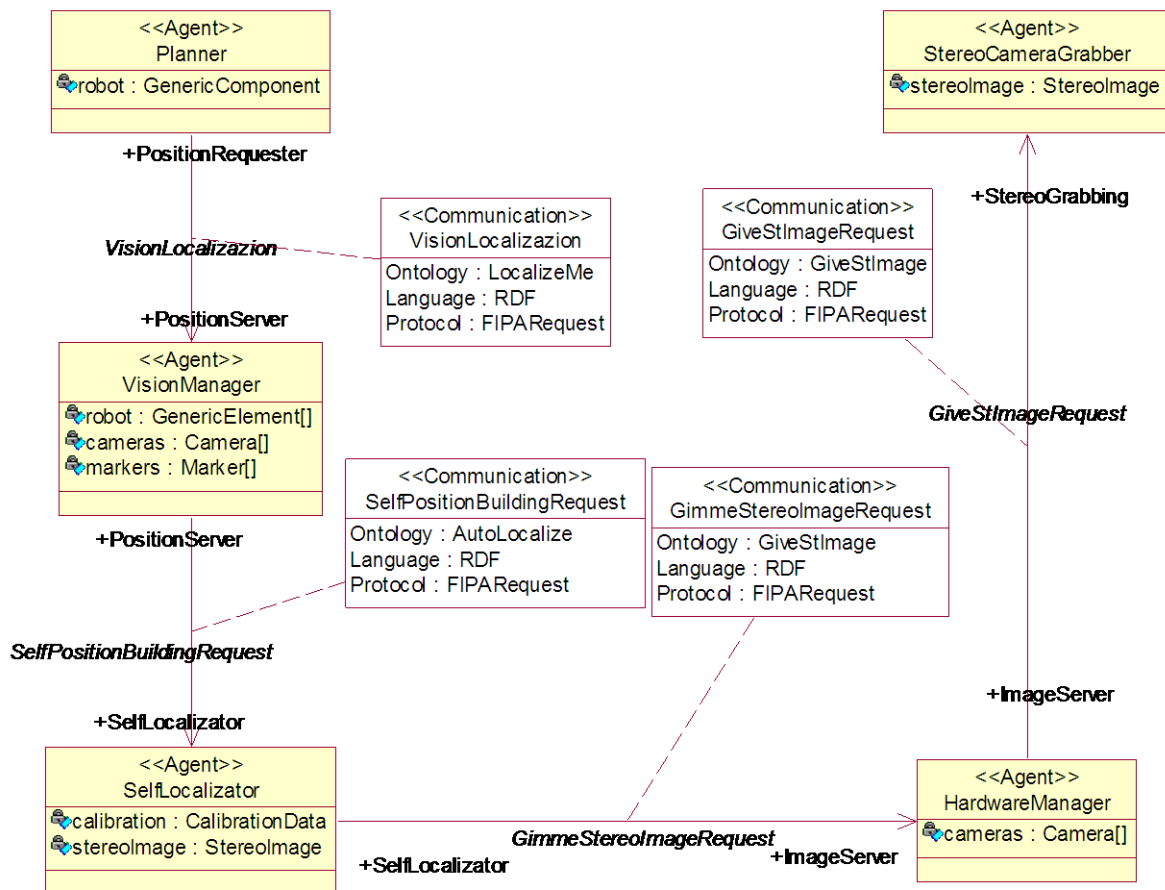


Figure12 : Diagramme de description de l'ontologie de communication

### 3.2. 3.la description des rôles :

L'objectif de ce fragment est de modéliser le cycle de vie de chaque agent, en examinant les rôles qu'il peut jouer, la collaboration dont il a besoin et les communications auxquelles il participe.

Le modèle UML de cette partie du processus, le diagramme de description des rôles, est conçu selon une notation UML standard.

Nous représentons le diagramme de description de rôle comme un diagramme de classes où les rôles sont des classes regroupées dans des packages représentant les agents.

Les rôles peuvent être connectés par des relations représentant des changements de rôle, des dépendances pour un service ou la disponibilité d'une ressource et des communications. Chaque rôle est obtenu en composant plusieurs tâches pour cette raison nous spécifions les tâches impliquées dans le rôle en utilisant le compartiment opération de chaque classe.

Plus en détails :

- Les classes représentent les rôles de l'agent. Ils sont regroupés dans des packages qui représentent l'agent.
- Les relations entre les rôles peuvent être de 3 types différents :
  - Communication. Représenté par une ligne continue dirigée de l'initiateur au participant. Les noms des communications proviennent du diagramme Description de l'ontologie de communication.
  - Dépendances. Comme dans  $i^*$ , nous pouvons avoir des dépendances de service ou de ressource. Une dépendance de service montre qu'un rôle dépend d'un autre pour atteindre un objectif (indiqué par une ligne pointillée avec le stéréotype de service). Dans la dépendance de ressource, un rôle dépend d'un autre pour la disponibilité d'une entité (indiqué par une ligne pointillée avec le stéréotype de ressource). Nous pouvons également avoir des dépendances de services logiciels et de ressources logicielles ; dans ce cas, le service/ressource demandé est utile ou souhaitable, mais pas essentiel pour atteindre l'objectif d'un rôle.
    - Changements de rôle. Cette connexion est décrite comme une relation de dépendance car nous voulons signifier la dépendance du second rôle vis-à-vis du premier. Parfois la condition de déclenchement n'est pas générée explicitement par le premier rôle mais son apparition précédente dans le scénario justifie la considération qu'il faut préparer la situation qui permet au second rôle de démarrer. Nous utilisons OCL ou du texte semi-formel pour exprimer la condition de déclenchement.

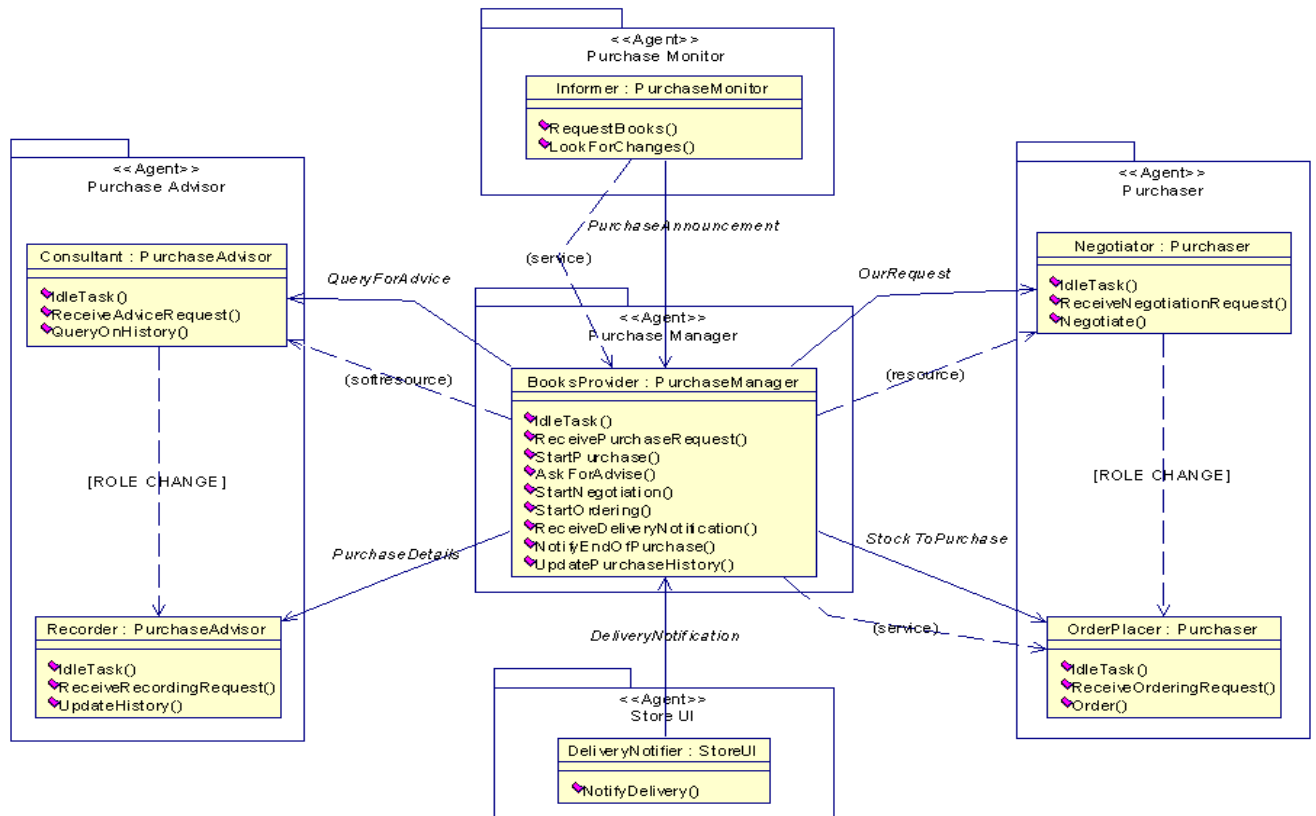


Figure 13 : Diagramme de description des rôles

**3.2.4. La description des protocoles de communication :** se fait grâce à des diagrammes de séquence ou de protocoles A-UML, suivant les standards FIPA par exemple.

Nous allons définir un fragment de méthode Protocol Description, extrait de la méthodologie PASSI dont le processus est complètement représenté dans la figure suivante. Ce fragment vise à représenter le protocole utilisé pour chaque communication tel que spécifié par l'architecture FIPA. Le modèle UML de cette partie du processus, la description du protocole, est conçu selon une notation AUML.

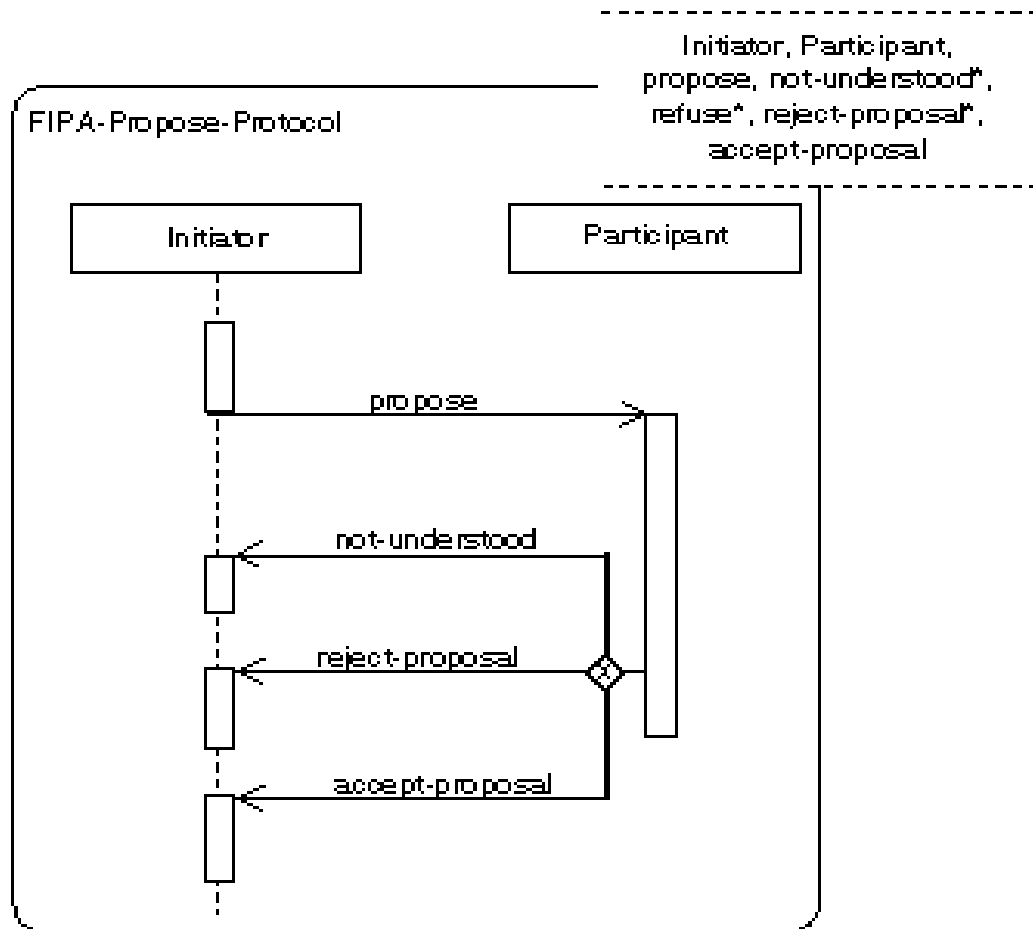


Figure 14 : Diagramme de séquence

### 3.3. L'implémentation des agents :

**3.3.1. La définition de la structure des agents :** permet de définir les constituants des agents en termes de ressources internes, de connaissances (attributs) et de tâches (méthodes) ;

Ce fragment vise à représenter l'architecture générale du système (les agents leurs connaissances et leurs tâches). Le modèle UML de cette partie du processus, le diagramme MASD, est conçu selon une notation UML standard

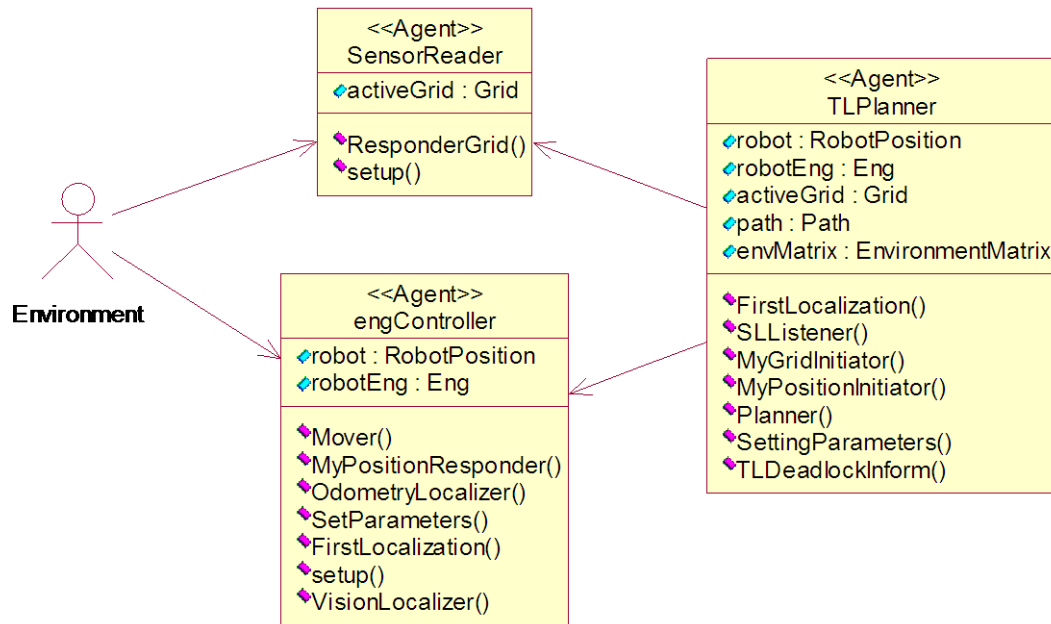


Figure 15 : Le diagramme MASD

Le diagramme de classes contient des classes et des acteurs. Chaque classe symbolisant un agent du système. Les compartiments d'attributs peuvent être utilisés pour représenter les connaissances de l'agent (en référence aux entités définies dans la description d'ontologie de domaine), tandis que les compartiments d'opérations sont utilisés pour signifier les tâches de l'agent. Les relations indiquent le flux d'informations échangées (communications)

**3.3.2 .La définition du comportement des agents :** est obtenue grâce à des diagrammes d'activité où apparaissent les tâches d'un agent et leurs interactions ;

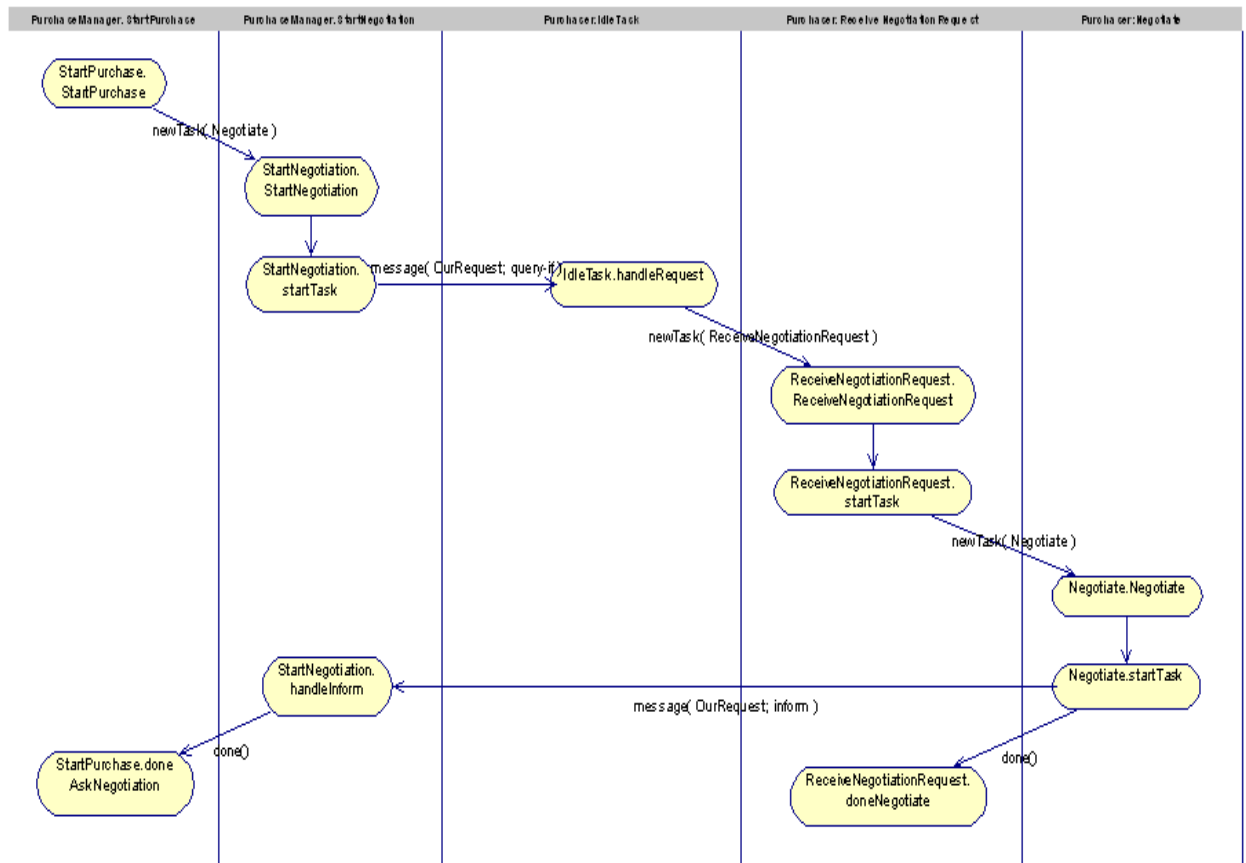


Figure 16 : Le diagramme MABD

Ce diagramme d'activité peut être utilisé pour montrer le flux d'événements entre et au sein des classes d'agents principaux et de leurs classes internes (représentant leurs tâches).

Nous utilisons un couloir pour chaque agent et pour chacune de ses tâches. Les activités à l'intérieur des couloirs indiquent les méthodes de la classe associée.

Les transitions habituelles de la norme UML sont décrites ici pour signifier soit des événements (par exemple, un message entrant ou une fin de tâche) soit l'invocation de méthodes.

Si la transition est liée à une conversation, le libellé signale le performatif et le contenu du message.

### 3.4. Le codage :

Elle est composée des activités suivantes :

**3.4. 1. La réutilisation du code :** peut être possible en cas de développements préexistants.

Nous allons définir un fragment de méthode Code Reuse Library, extrait de la méthodologie PASSI dont le processus est complètement représenté dans la figure suivante ;

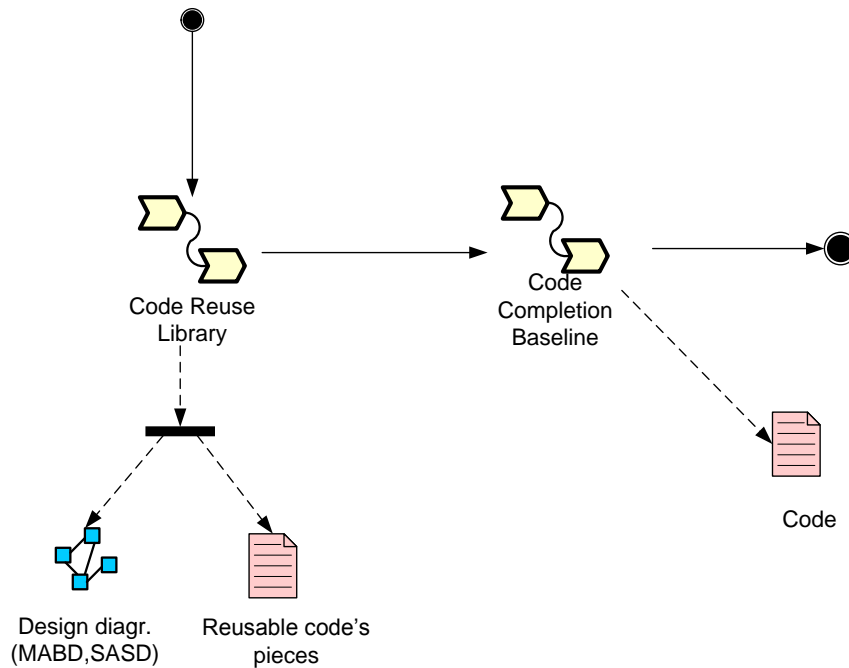


Figure 17 : La phase de code

Considérons la définition de travail "Code Reuse Library" dont le but est d'essayer de réutiliser des modèles d'agents et de tâches existants.

Le processus à effectuer pour obtenir le résultat est représenté sur la fig. 3 sous forme de diagramme SPEM

**3.4. 2. La complétion du code :** peut se faire à travers des outils dédiés et permet de rendre exécutable le code des agents.

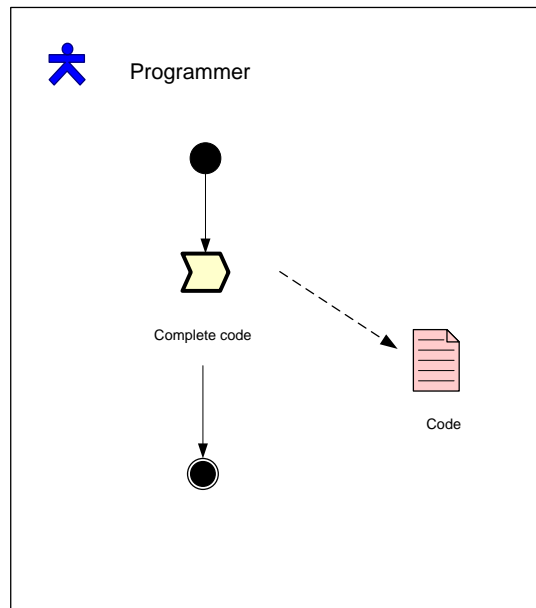


Figure 18 : Achèvement du code Fragment de référence - Aspect procédural

Il s'agit plutôt d'une phase conventionnelle. Le programmeur complète le code de l'application à partir du design, du squelette produit et des motifs réutilisés.

Il s'agit plutôt d'une phase conventionnelle. Le programmeur complète le code de l'application à partir du design, du squelette produit et des motifs réutilisés.

### 3.5. Le deployment:

Considérez le processus PASSI et la phase agent implémentation avec son résultat “Agent implémentation model”, l'ordre des activités effectuées dans ce fragment est montré dans le diagramme SPEM suivant le fragment décrit ou se trouvent les agents et de quelles différentes unités d'élaboration ils ont besoin afin de communiquer entre eux.

**3.5.1. la configuration du déploiement :** est nécessaire lorsque l'on spécifie des agents mobiles. Cette étape permet de définir la topologie du réseau et l'attribution/permission d'accès aux ressources pour les agents. des besoins au déploiement, les étapes de PASSI sont très clairement définies ainsi que les modelés à fournir tout au long du processus .PASSI est une méthode pouvant s'appliquer à n'importe quel domaine, mais est plutôt orientée agent mobile, avec notamment les aspects respectueux des spécifications FIPA. LE manque de formalisme, cependant, rend les phases de tests peu efficaces

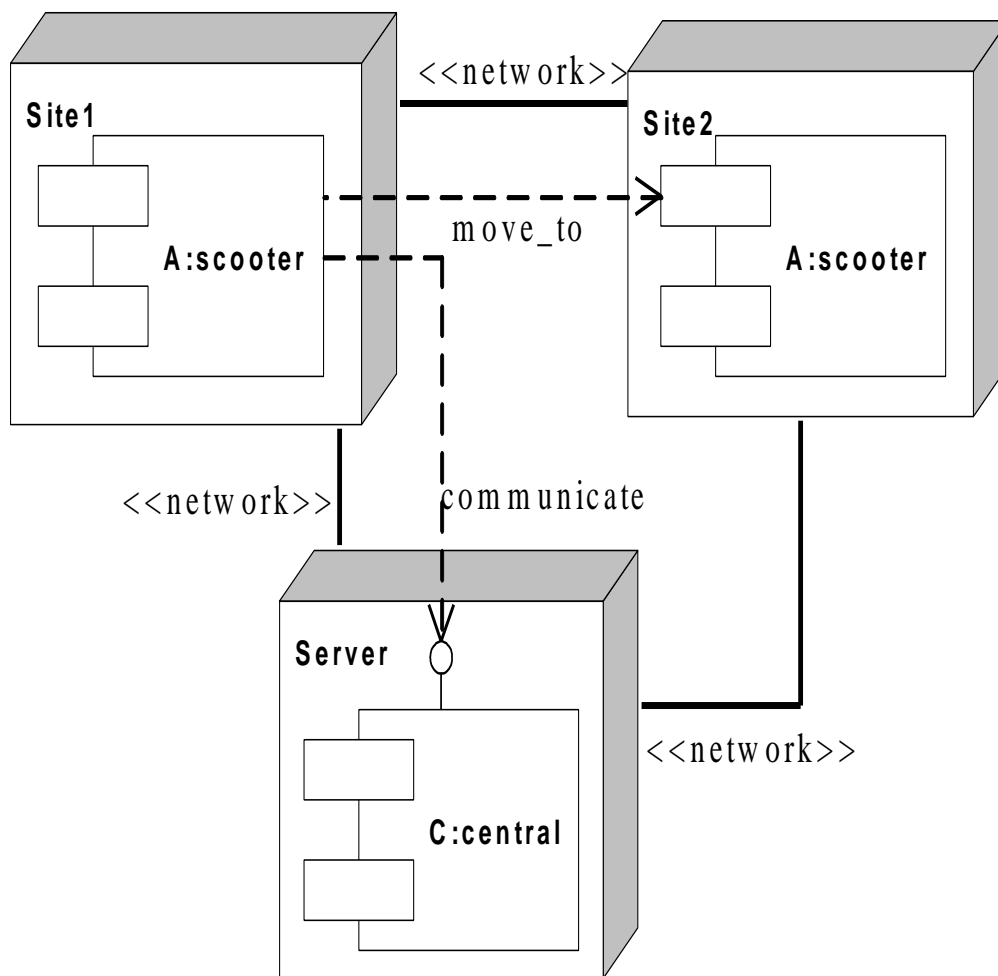


Figure 19 : The Déploiement Configuration diagramme

#### 4. Conclusion :

PASSI est une méthodologie qui couvre en grande partie le processus de développement d'un SMA. Elle permet de suivre l'évolution du SMA depuis la définition des objectifs jusqu'à la conception et l'implémentation final de l'application.

De plus, Cette méthodologie apporte une dimension organisationnelle au SMA grâce à la définition de rôles précis pour chaque agent du système, une caractéristique importante dans notre cas. Enfin, au vu des nombreuses entrées/sorties pouvant avoir lieu dans le système que nous concevons, nous nous devons de choisir une méthodologie qui prend en compte cet aspect dynamique.

# Chapitre 03 : Analyse et conception

## 1. Introduction :

Les systèmes multi-agents (SMA) ont montré leur pertinence pour la conception d'application distribuée (logiquement ou physiquement), complexes et robustes. Le concept d'agent est aujourd'hui plus qu'une technologie efficace, il représente un nouveau paradigme pour le développement de logiciels dans lesquels l'agent est un logiciel autonome qui possède un objectif, évolue dans un environnement dynamique et interagit avec d'autres agents au moyen de langages et de protocoles. Souvent, l'agent est considéré comme un objet « intelligent » ou comme un niveau d'abstraction au-dessus des objets et des composants.

L'objectif de cet article est d'établir la spécificité du paradigme multi-agent.

Plusieurs méthodes sont proposées pour modéliser ce type de système tels que Passi, Mase, Tropos, Gaia, Prometheus, ADELFE et Message .Dans ce chapitre nous avons choisi la méthode Passi qui va nous aider a la conception de notre système.

## **2. Conception avec la méthode Passi :**

Est une méthode pas a pas intégrant des modèles de conception et concepts provenant de l'ingénierie orientée objet et de l'intelligence artificielle en utilisant UML. le méta-modèle d'agent de PASSI manipule beaucoup de concepts communs aux agents et y ajoute les notions provenant des contraintes FIPA pour être adapté a FIPAOS ou JADE. Ces notions sont encapsulées dans cinq modèles les besoins, la définition de la société d'agents, modèle de mise en œuvre agent (l'implémentation des agents) modèle de codage, modèle de déploiement. Chaque phase est composée d'étapes, correspondant aux spécifications des cinq modèles présentés précédemment.

## 3. Spécification des besoins :

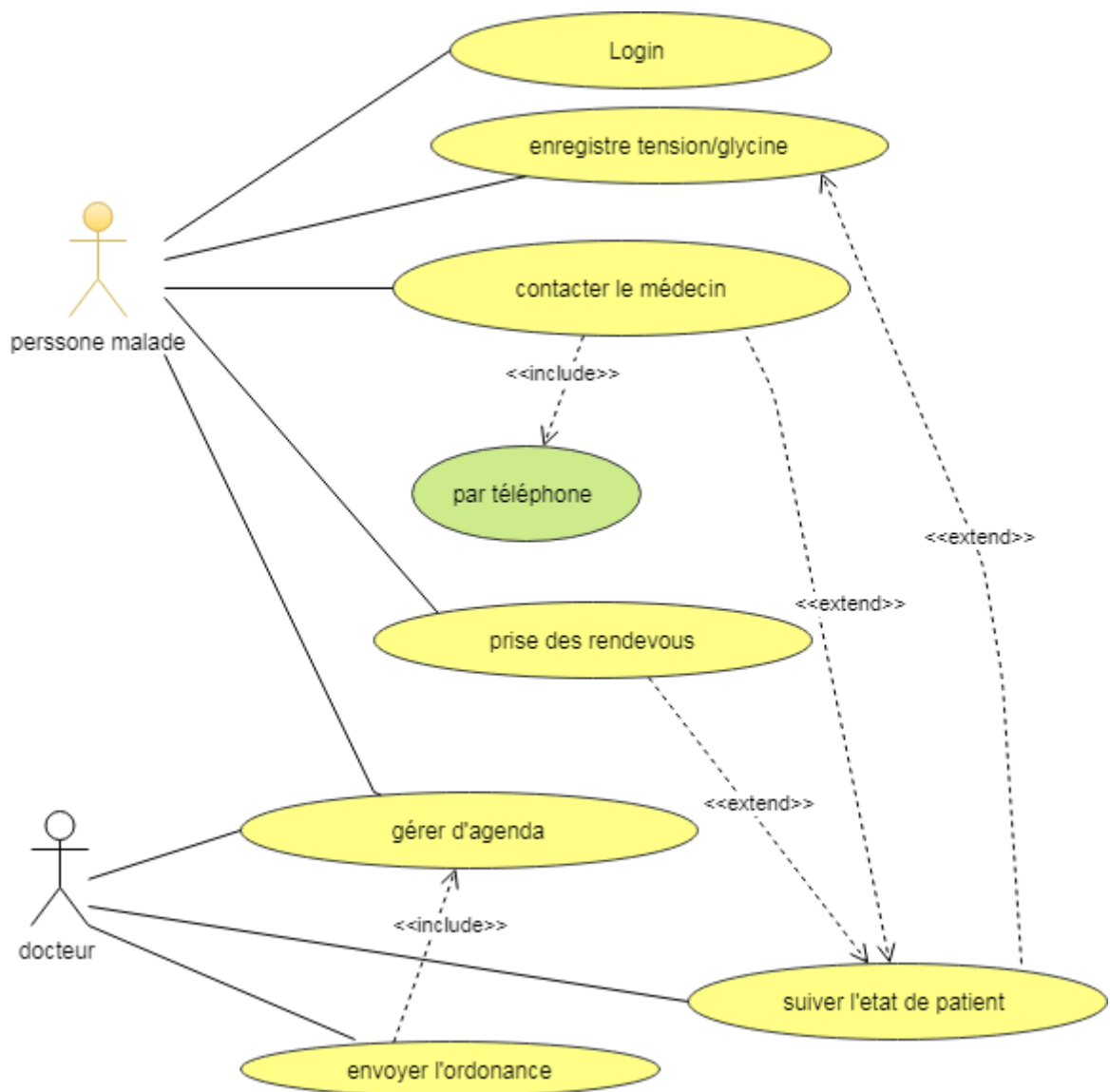
La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins de notre application. Nous distinguons des besoins fonctionnels qui présentent les fonctionnalités attendues de notre application et les besoins non fonctionnels pour éviter le développement d'une application non satisfaisante ainsi de trouver un accord commun entre les spécialistes et les utilisateurs pour réussir le projet.

Après une étude détaillée de système, cette partie est réservée à la description des exigences des différents utilisateurs de l'application. Ces besoins se regroupent dans les modèles des besoins. Les besoins d'utilisateur :

- ✓ Login
- ✓ Prendre rendez-vous
- ✓ Contacter le médecin par téléphone
- ✓ Suivre l'état patient
- ✓ Gestion d'agenda
- ✓ Enregistrer tension/glycémie
- ✓ Envoyer chez médecin

### **3. Tâche 1 :**

La description du domaine : permet d'exprimer les besoins et de décrire le contexte du système grâce à une hiérarchie de cas d'utilisation ;



*Figure 20: diagramme de description du domaine*

**4. Tâche 2** : L'identification des agents : est une étape peu présente dans les méthodes orientées agent, et est effectuée par regroupement en paquetages, qui correspondront aux agents ;

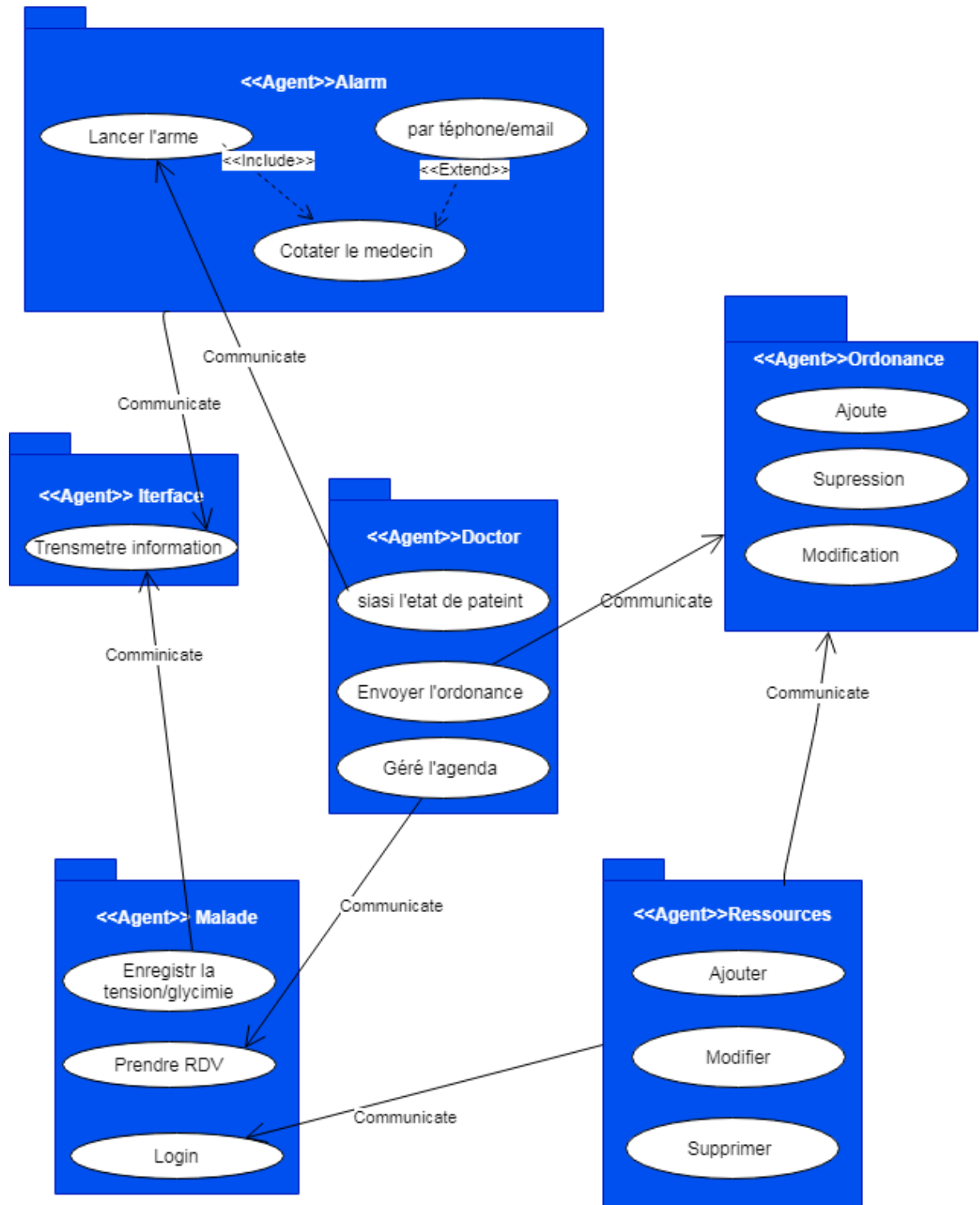


Figure 21: diagramme d'identification des agents

**5. Tâche 3 :**

L'identification des rôles : Les diagrammes de séquence décrivent toutes les interactions possibles entre les agents. Ces interactions décrivent un scénario de travail d'agent interagissant pour obtenir un comportement souhaité du système. Les rôles sont à la fois une description fonctionnelle et comportementale des agents, mais aussi une représentation des relations entre agents. Chaque agent peut appartenir à plusieurs scénarios, qui sont établis au moyen de diagrammes de séquence ;

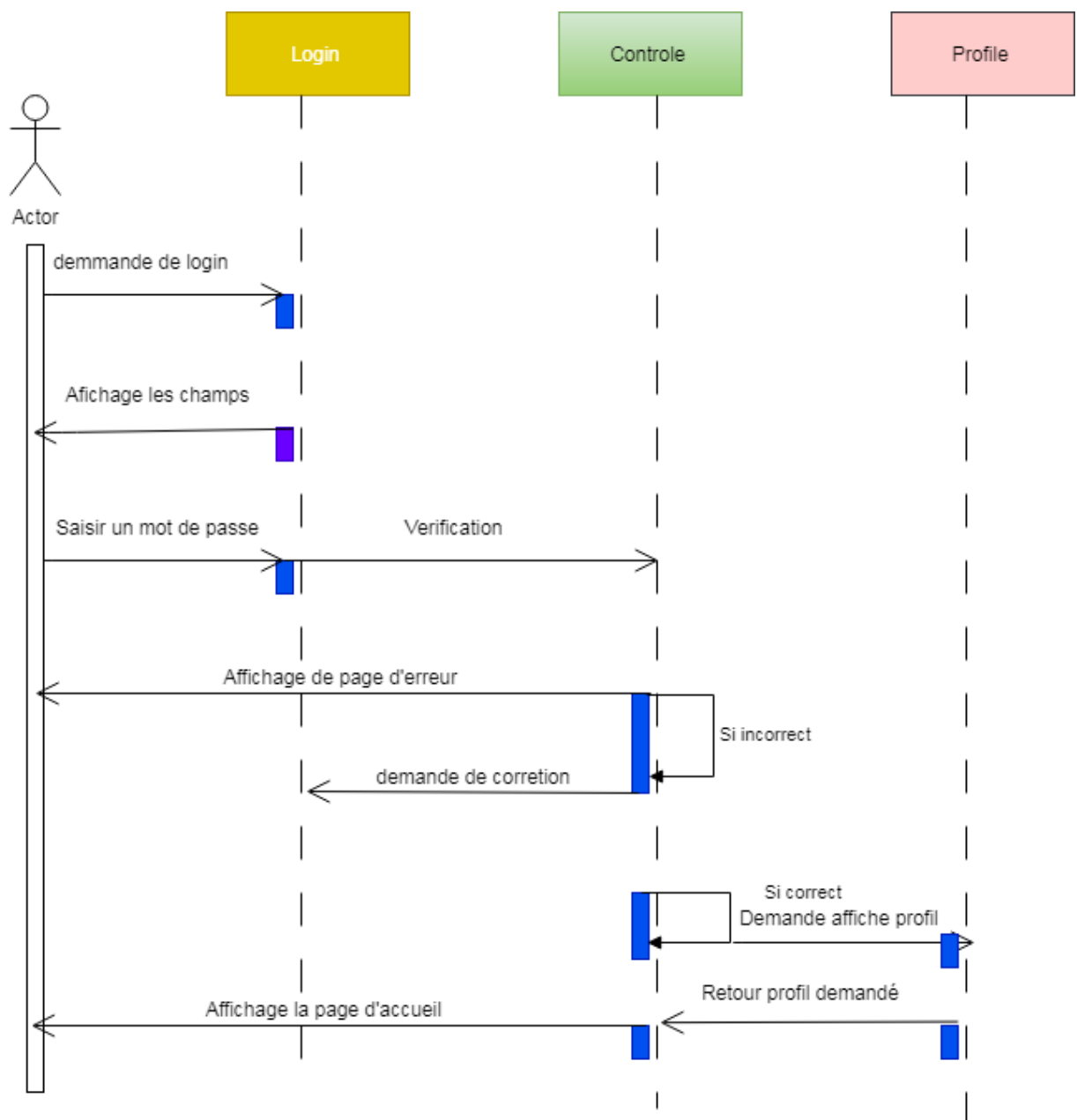


Figure 22: diagramme d'identification des rôles pour « Login »

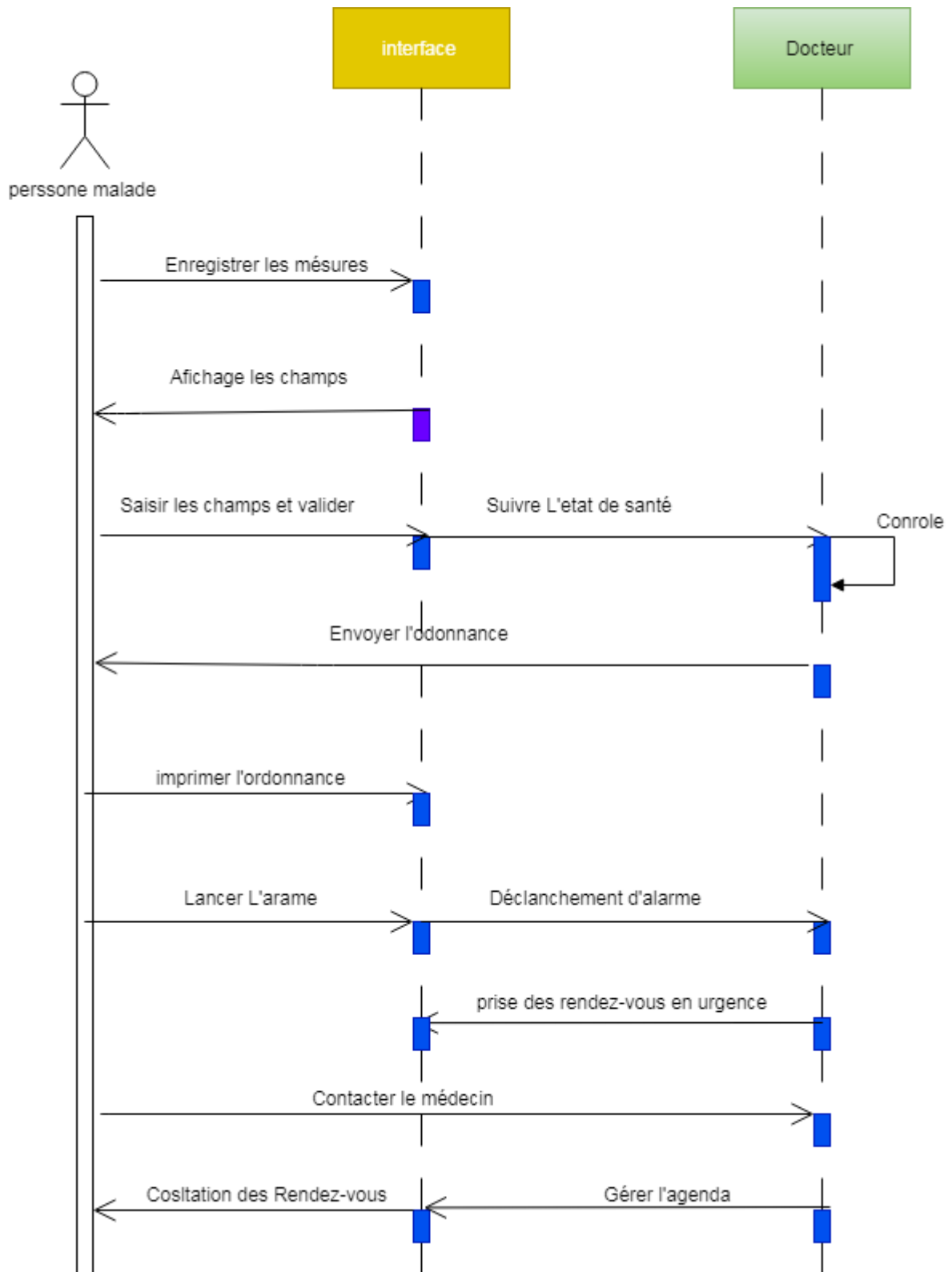


Figure 23: diagramme d'identification des rôles pour « système »

**6. Tâche 4 :**

La spécification des tâches : permet de définir les activités des agents en les reliant aux fonctionnalités fournies par le système ;

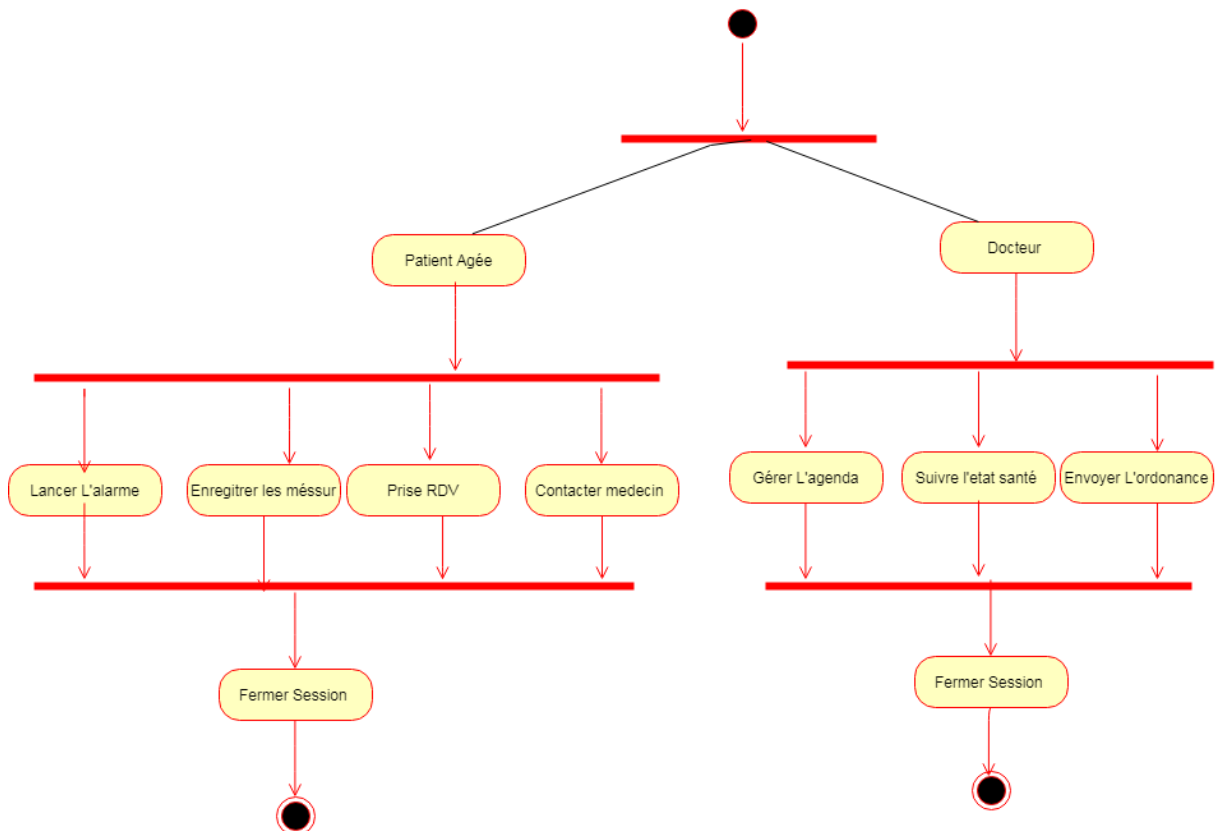


Figure 24 : diagramme de spécification des tâches

**7. Conclusion :**

Ce chapitre a donné une vision sur travail et a donné l'aspect conceptuel de l'application à travers les différents diagrammes décrits en PASSI. Ce chapitre qui suit fera l'objet de la mise en œuvre de notre application.

# Chapitre 04 : Implémentation

## 1. Introduction :

Après avoir établi une étude conceptuelle de notre application, nous passons à l'implémentation de cette dernière définis et détaillée au chapitre précédent. Pour réaliser notre travail, nous avons choisi le langage de programmation Java en utilisant la plate-forme JADE.

## 2. Présentation du langage Java :

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes modernes et performants, des jeux et des applications professionnelles. Java est utilisée sur plus de 850 millions d'ordinateur de bureau et plus d'un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle.

## 3. Les caractéristiques de java :

Il est donné de nombreuses fonctionnalités de java .Les caractéristiques Java ci-dessous sont simples et faciles à comprendre ;

- **Java est portable :** nous pouvons le byte code java pour toute plateforme.
- **Java est orienté objet :** orienté objet signifie que nous organisons notre logiciel comme une combinaison de différents types d'objets qui intègre à la fois la programmation orienté objet (OOP) est une méthodologie qui simplifient le développement et la maintenance de logiciels en fournissant des règles.
- **Java est simple :** le langage java est simple parce que la syntaxe est basée sur C++ (donc plus facile pour les programmeurs d'apprendre après C++).
- **Distribué :** nous pouvons créer des applications distribuées en java. RMI et EJB sont utilisés pour créer des applications distribuées. Nous pouvons accéder à des fichiers en appelant les méthodes de toute machine sur l'internet.
- **Robuste**
- **Haute performance**
- **Independent de la plate-forme :** code java peut être exécuté sue de multiples plates-formes ex : Windows, Linux, le code Mac OS...java est compilé par le compilateur et converti en Byte code.
- **Architecture-neutre :** il n'y a pas, par exemple caractéristiques dépendent taille des types primitifs de mis en œuvre est réglé.

**•Sécurisé :**

- ✓ **Byte code vérifier**
- ✓ **Class loader**
- ✓ **Sécurité manager**

**4. Présentation de la plate-forme JADE :****Définitions****Définition1 :**

JADE (Java Agent Développement Framework) est une plate-forme multi-agent créé par le laboratoire TILIB. JADE permet le développement de système multi-agents et d'applications conformes aux normes FIPA. Elle est implémentée en JAVA et fourni des classes qui implémentent <<JESS >>pour la définition du comportement des agents.

**Définition2 :**

Est un projet Open Source, LGPL License Contrôlée par Telecom Italie Lab. qui reste propriétaire du projet. Il existe deux produits principaux :

- Plate-forme agent satisfaisant aux spécifications de la FIPA.
- Api pour développer des agents en java

JADE possède trois modules principaux (nécessaire aux normes FIPA).

- DF<< Director Facilitor >> fournit un service de <<pages jaunes>> à la plateforme ;
- ACC<<Agent Communication Channel>> gère la communication entre les agents ;
- AMS<<Agent Management System>> supervise l'enregistrement des agents, leur authentification, leur accès et l'utilisation du système.

Ces trois modules sont activés à chaque démarrage de la plate-forme.

**5. Caractéristiques :**

- Support à l'envoi de messages, transparent et multi-protocole**
- Diffusion d'événements en local**
- Java RMI pour la diffusion interne à une plateforme**
- FIPA 2000 Message Transport Protocol :**
  - ✓ protocole IIOP pour la diffusion inter-plateforme.
  - ✓ protocole HTTP et encodage des ACL en XML
- Modèle de concurrence à deux niveaux**

- ✓ Entre agent (pre-emptif, Threads Java).
- ✓ Interne aux agents (coopératif, classes de comportements “behaviour”).

#### • Mobilité des agents

- ✓ Entre plateforme, mobilité faible.
- ✓ Framework orienté objet implémentant en Java les spécifications FIPA.
- ✓ Plateforme agent.
- ✓ Langage de communication agent.
- ✓ Protocoles d’interaction standard.

#### • Système de gestion d’événements dans le noyau de la plateforme

- ✓ Permet l’observation de la plateforme, des messages, du transport des messages, des agents.

#### • Agents utilitaires

- ✓ Dum myAgent tool permet à des utilisateurs d’interagir avec les agents déployés sur la plateforme.
- ✓ Sniffer Agent : agent utilisé pour observer les messages.

### 6. Plateforme JADE :

- ✓ Une application JADE est une plateforme déployée sur une ou plusieurs machines.
  - ✓ La plateforme héberge un ensemble d’agents, identifiés de manière unique, pouvant communiquer de manière bidirectionnelle avec les autres agents.
  - ✓ Chaque agent s’exécute dans un conteneur (container) qui lui fournit son environnement d’exécution ; il peut migrer à l’intérieur de la plateforme.
  - ✓ Toute plateforme doit avoir un conteneur principal qui enregistre les autres conteneurs.
- Une plateforme est un ensemble de conteneurs actifs.

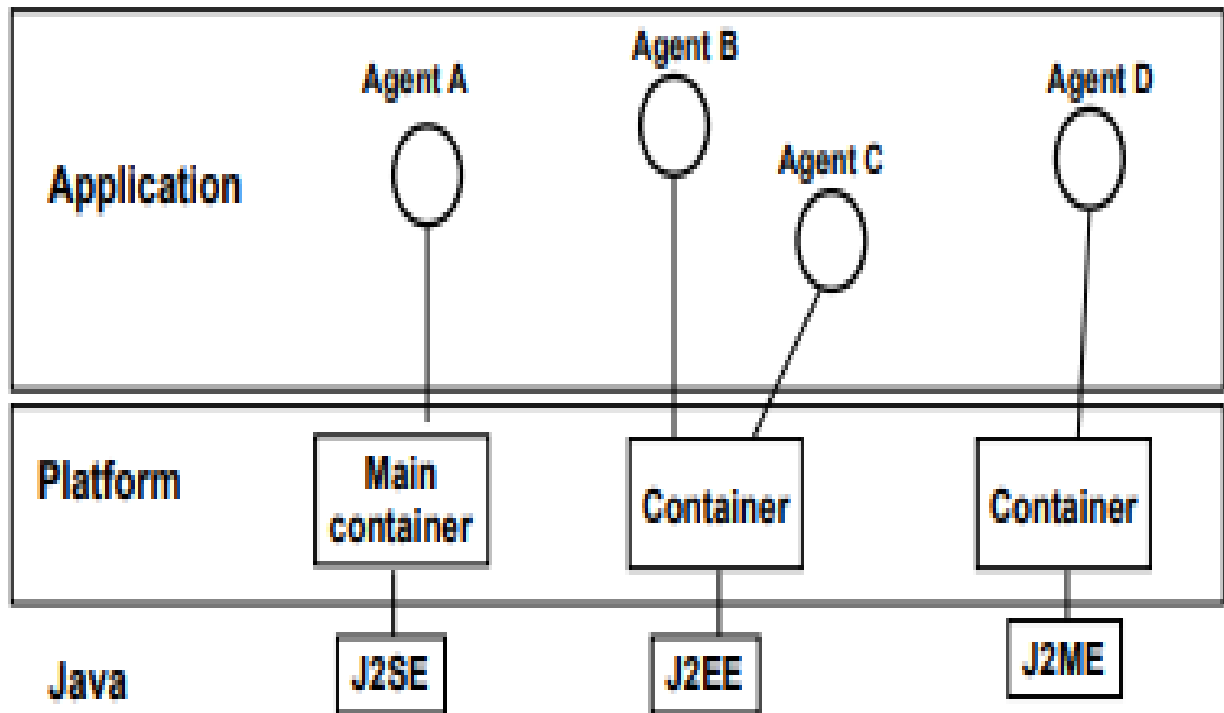


Figure 25 : La plate-forme JADE

### 7. Conteneurs :

- Un conteneur est un environnement d'exécution JADE :
- Environnement complet d'exécution pour un agent
- Exécution concurrente de plusieurs agents
- Contrôle le cycle de vie des agents
- Assure la communication entre agents

### 8. Outils utiles au débogage

#### •Dum my Agent

- ✓ Visualisation des messages
- ✓ Envoie des messages aux agents présents sur la plateforme et réceptionne leur réponse.

#### •Sniffer Agent

- ✓ Visualisation de l'enchaînement des messages et des messages eux même.
- ✓ Vérification interactive de la correction des protocoles.

#### •Introspector Agent

- ✓ Visualisation des messages envoyés/reçus,
- ✓ Visualisation des Behaviours actifs et non actif,
- ✓ Contrôle de l'état de l'agent

### 9. Architecture de la plateforme Jade

Dans la plate-forme JADE, deux méthodes fournies par la classe Agent afin d'obtenir l'identifiant de l'agent DF par défaut et celui de l'agent ASM : `getDefaultDF ()`. Ces deux agents permettent de maintenir une liste des services et des adresses de tous les autres agents de la plate-forme. Le service DF propose quatre méthodes afin de pouvoir :

- Enregistrer un agent dans les pages jaunes.
- Supprimer un agent des pages jaunes.
- Modifier le nom d'un service fourni un agent.
- Rechercher un service.

Le service AMS s'utilise généralement de manière transparente (chaque agent créé est automatiquement enregistré de l'AMS et se voit attribué une adresse unique).

Ces deux services fournissent donc les annuaires qui permettent à n'importe quel agent de trouver un service ou un autre agent de plate-forme.

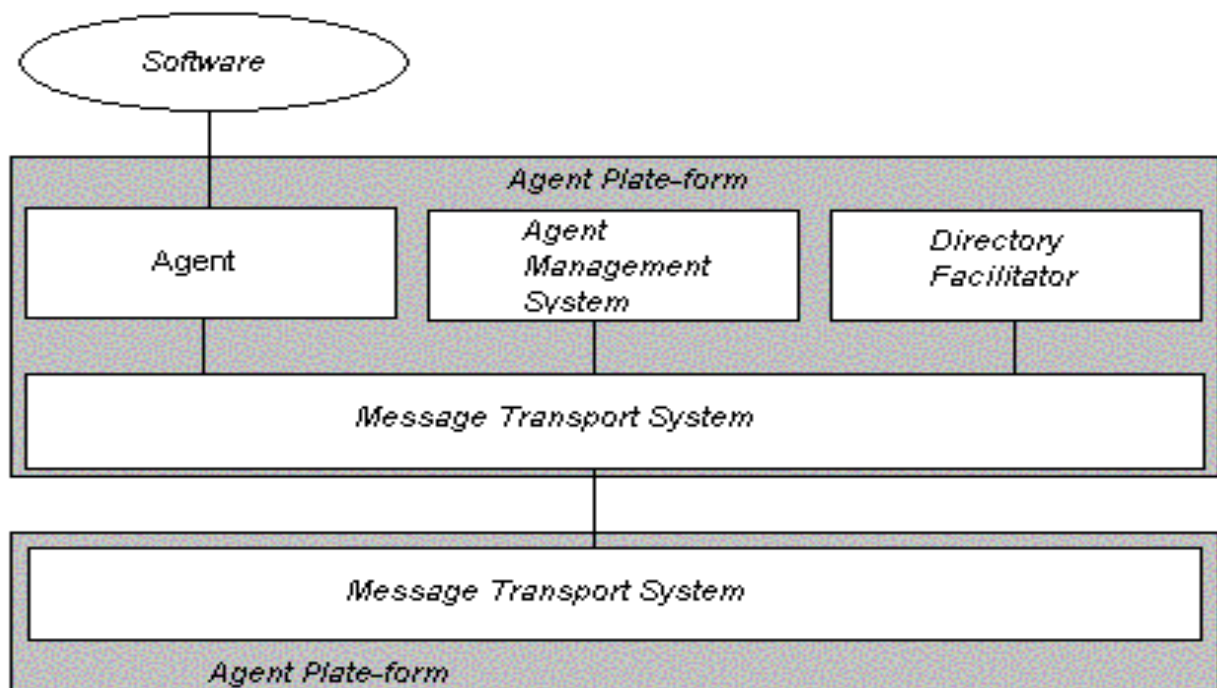


Figure 26 : Architecture logiciel de la plate-forme JADE

## 10. L'interface graphique de JADE :

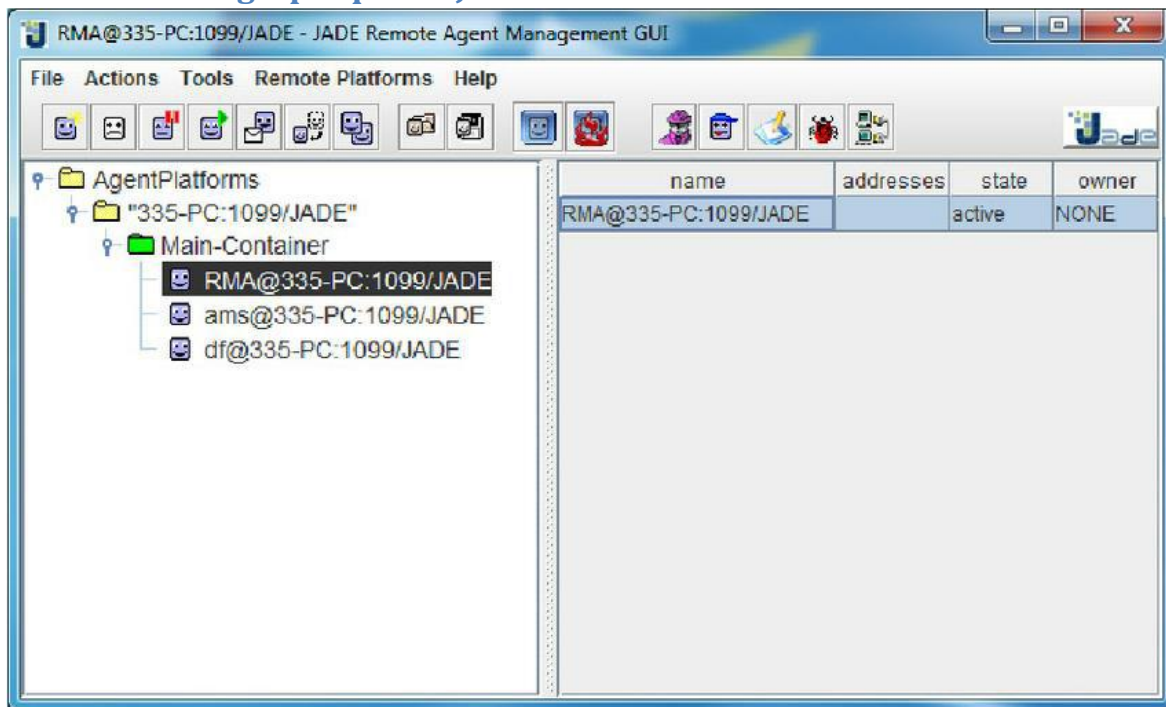


Figure 27 : Interface graphique de JADE

## 11. Installation et configuration :

### •Installation :

- ✓ Source disponibles à <http://jade.tilab.com>

### •Configuration :

- ✓ Ajouter dans le classpath les archives java (.jar) se trouvant le répertoire lib
- ✓ Archives nécessaires :
  - Jade.jar.
  - JadeTools.jar.
- ✓ Archives optionnelles en fonction du type de communication inter plate-forme :
  - Hhttp.jar.
  - Iiop.jar.

## 12. Lancement de la plateforme :

### •Lacement de base :

- ✓ Java jade. Boot [liste agents].

### •Lancement avec interface graphique :

- ✓ Java jade. Boot-gui [liste agents].

- Note** : l'interface graphique est représentée par le Remote Monitoring Agent (RMA).

### 13. Lancement d'agents :

#### •Lancer d'agents :

✓ Java jade. Boot my Agent: my Package.myAgent.

•**Note:** il est nécessaire de préfixer le nom de la class de l'agent par un nom.

### 14. Captures d'écran sur quelques interfaces de l'application :



Figure 28: Interface département (Société)



Figure 29: Interface Design Preview (login)

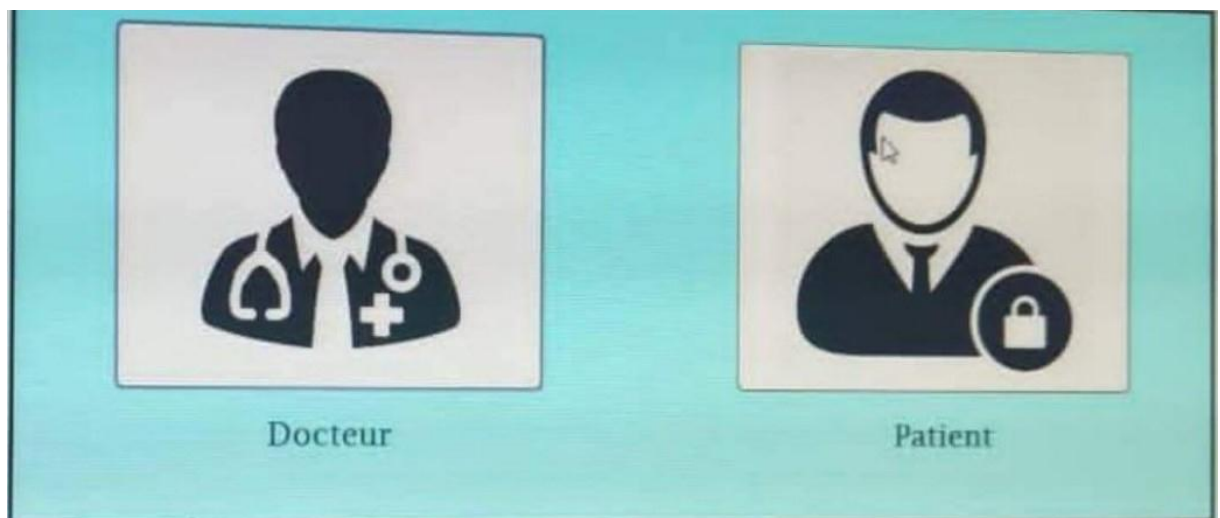


Figure 30: Interface Authentication (Docteur/Patient)



Figure 31: Interface Accueil pour patient



Figure 32: Interface accueil médecin

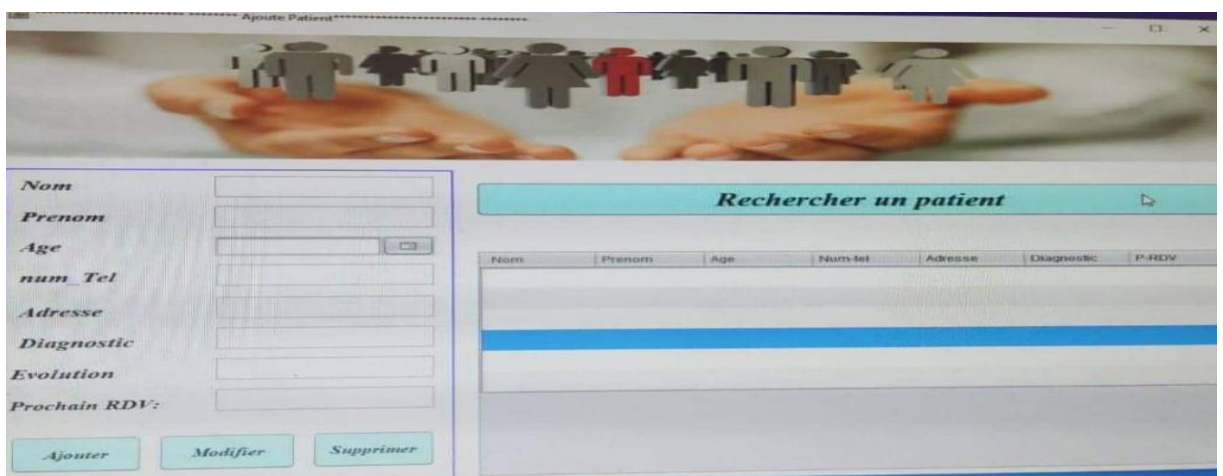


Figure 33 : Interface ajout/ recherche des patients

The screenshot shows a web application interface with the following elements:

- Header: Three tabs labeled "coordonnées", "departement", and "contrat de travail".
- Banner: An image showing two hands holding a row of stylized human figures, with one figure in red.
- Form Fields (left side):
  - Num-sociale:
  - Nom:
  - Prenom:
  - Age:  (with a calendar icon)
  - Civilité:  (dropdown menu)
  - Situation familiale:  (dropdown menu)
  - num\_Tel:
  - Adresse:
- Buttons: "Ajouter", "Modifier", and "Supprimer" (light blue buttons).
- Table (right side):
  - Table header: Num-social, Nom, Prenom, Age, Civilité, Situation, Num-tel, Adresse.
  - Table body: A large empty grey area, likely representing a table with no data rows.
- Buttons: "Actualiser" (grey button) and "Actualiser" (grey button) are located above the table.

Figure 34 : Interface d'enregistrement des mesures

### **15. Conclusion :**

A travers ce chapitre, nous avons présenté les outils utilisés durant la phase d'implémentation du système d'agents représentant notre application de maintien à domicile des personnes âgées, ainsi que l'application réalisée et les multiples scénarios possibles durant l'exécution de l'application.

# Conclusion générale

Un grand nombre de clients peuvent bénéficier du maintien à domicile pour une multitude de raisons. Ce service peut être destiné aux personnes âgées, aux handicapés et aux personnes malades. L'objectif est d'accompagner les gestes du quotidien afin de s'assurer que ceux-ci sont bien réalisés et en toute hygiène, tout en maintenant au maximum l'autonomie de la personne aidée. Ainsi, les patients auront les soins indispensables à leur domicile. L'aide à domicile est également fournie aux personnes souffrant de maladies

En choisissant le maintien domicile des personnes âgées peuvent conserver leur indépendance. Les recherches montrent que c'est un facteur important de la santé physique et mentale à long terme. Le maintien à domicile aide les personnes âgées à conserver leur dignité. Un assistant de soins à domicile dispose des compétences nécessaires pour les épauler dans l'accomplissement de ces tâches tout en apportant le plus grand soin à respecter l'intimité de la personne à aider.

Les services de maintien à domicile peuvent être personnalisés. Les clients et les familles peuvent s'attendre à ce que leurs besoins uniques soient satisfaits, l'aide étant différente pour chaque pathologie ou encore âge de la personne en situation de perte d'autonomie. L'aide à domicile peut intervenir à toute heure du jour et de la nuit (garde de jour et garde de nuit). C'est un soutien fiable qui inspire confiance. Le maintien à domicile réduit le niveau d'inquiétude de la famille et apporte une importante tranquillité à l'esprit. Les familles pouvant se trouver loin de leurs proches âgés, une aide à domicile pourra les informer de l'état de santé de la personne aidée.

Même si le maintien à domicile propose de nombreux avantages, il peut aussi avoir quelques inconvénients.

Le choix de service de soins peut être réduit selon le lieu d'habitation. Si la personne ayant besoin d'un service d'aide à domicile vit dans un village isolé, cela peut représenter un temps de trajet important pour l'aide à domicile et l'obligation d'avoir un véhicule pour venir sur place. Il faudra par ailleurs, malgré le plus faible nombre de professionnels présents dans ce contexte, que l'aide à domicile convienne à la personne aidée, l'assistance étant au-delà des actes du quotidien une présence humaine et chaleureuse.

## Conclusion générale

---

Bien que le maintien à domicile soit moins cher, en cas de dépendance trop importante et pathologies nécessitant une garde 24/24, la résidence pour personnes âgées peut apporter le matériel adapté comme des rampes, des garde-corps ou des télésièges à ses habitants.

# Références

- [1]: Wikipédia <https://fr.wikipedia.org> › wiki ›
- [2]: J.-P. D. JAMONT, *Une approche pour la conception de systèmes multi-agents embarqués*, Grenoble: Thèse de doctorat, 2005.
- [3]: M. WEISER, «The computer for the 21st century, » chez *Scientific American*, 1991.
- [4]: Do-It-Yourself » d'habitats intelligent - Saviors UdeS  
<https://savoirs.usherbrooke.ca> ›
- [5]: Wikipédia <https://fr.wikipedia.org> › wiki ›
- [6]: K. Curran, «Ambient Intelligence: the hidden natural intelligence, » *International Journal of Ambient Computing and Intelligence*, p. ii, 2009.
- [7] : J.-M. Favre, J. Estublier et M. Blay Forna\_rino : L'ingénierie dirigée par les modèles : au-delà du MDA ; série Informatique et Systèmes d'Information, Hermès Sciences Lavoisier, janvier 2006.
- [8]: A. Perini ET A. Susi: Automating model transformations in agent-oriented model ling; 6th International Workshop on Agent Oriented Software Engineering (AOSE'05), Utrecht, NL, 25-26 juillet, 2005
- [9]: N. Gaud : Systèmes multi-agents holoniques : de l'analyse à l'implantation ; Thèse de l'Université de Franche-Comté et de l'Université de Technologie de Belfort-Montbé\_liard, décembre 2007
- [10]: C. Bernon, M. Cossentino, M.-P. Gleizes, P. Turci et F. Zambonelli : A study of some multi-agent meta-models ; Fifth International Workshop on Agent-Oriented Software Engineering (AOSE'04) at AAMAS'04, New York, États-Unis, juillet 2004, P. Giorgini, J. Mueller, J. Odell (Eds.), Springer Verlag, LNCS 3382, pp. 62-77, 2005.
- [11]: M. Cossentino, S. Gaglio, A. Garro ET V. Seidita: Method fragments for agent design methodologies: from standardisation to W MULTI-AGENTS ET GÉNIE LOGICIEL Figure 3. Génération d'API dédiée dans ADELFE GÉNIE LOGICIEL Q N o 86 SEPTEMBRE 2008 7 research; *International Journal of Agent Oriented Software Engineering*, 1(1), pp.91- 121, avril 2007.
- [12]: B. Henderson-Sellers: Creating a comprehensive agent-oriented methodology: Using method engineering and the OPEN metamo\_del; dans [18], pp. 368-398, 2005
- [13]: Cossentino, 2001; Cossentino ET Potts, 2002.

