

République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université du 20 Août 1955 SKIKDA

Faculté des sciences

Département d'informatique



Thèse

En vue de l'obtention du diplôme de

Doctorat de 3ème cycle (LMD) en Informatique

Option : Computation et Cognition des Systèmes Informatiques

Intitulé :

**Intelligence Ambiante et Environnements
Smart pour Personnes à Besoins
Spécifiques**

Présentée par :

ASMA BOUAITA

Membres du jury :

Président : Dr. Djamel Zeghida

Examinatrice : Pr. Nora Bounour

Examineur : Dr. Sofiane Boulehouache

Encadrant :

Pr. Smaine Mazouzi

Co- Encadreur

Dr. Yacine Kissoum

REMERCIEMENTS

D'abord, je remercie Dieu le Tout-Puissant de m'avoir donné la force, la patience et la volonté d'arriver au terme de ce travail.

Je remercie vivement mon directeur de thèse Monsieur Smaine MAZOUZI, Professeur à l'Université 20 Août 1955-Skikda pour son aide et ses conseils judicieux et constructifs, ainsi que sa disponibilité permanente et son soutien moral.

Je tiens à exprimer ma profonde gratitude à mon co-directeur de thèse Monsieur Yacine KISSOUM, Maître de conférences à l'Université 20 Août 1955-Skikda pour son encadrement et pour l'encouragement et l'intérêt qu'il m'a apporté pour l'accomplissement de ce travail et surtout pour sa grande aide et ses qualités humaines.

Mes remerciements vont également Aux membres du jury : Monsieur Djamel Zeghida, Madame Nora Bounour et Monsieur Sofiane Boulehouache, pour avoir consacré leur temps et leur expertise à l'évaluation de ce travail de recherche. Leur expertise et leurs remarques enrichiront indéniablement la portée de cette recherche.

Un remerciement particulier à Ma Mère et Mon Père qui, par leurs encouragements inconditionnels, m'ont permis d'atteindre ce but.

Je voudrais remercier vivement mon mari, mes sœurs, mes frères, mes nièces et neveux, tous mes amis, pour leur soutien moral très précieux et inestimable tout au long de ces années.

Merci

Dédicace

*Avec tout l'amour que l'on peut porter dans nos cœurs, je dédie ce travail aux deux êtres qui me sont les plus chers au monde : **MES PARENTS**,*

*Pour m'avoir comblée d'amour et d'affection, pour m'avoir soutenue et encouragée dans tout ce que j'ai entrepris, et pour avoir fait de moi la personne que je suis aujourd'hui, je leur exprime ma **GRATITUDE ÉTERNELLE**.*

*À mon cher mari, **NADJIB**,*

Merci pour ta patience, ton soutien indéfectible et ta présence rassurante à chaque étape de ce parcours.

*À mes enfants : **ISRAA, RACIM et GHAIH**,*

Que Dieu vous protège. Je vous aime plus que tout.

*À mes très chers frères : **HALIM et HOCINE**,*

*À mes très chères sœurs : **NADIA, KENZA et AICHA**,*

À mes nièces et neveux,

Merci pour votre amour et vos encouragements constants.

À toutes mes amies,

Que Dieu vous récompense pour tous vos bienfaits, vos mots d'encouragement et votre bienveillance tout au long de ce chemin.

ASMA

Résumé

Prendre soin de ses parents devient, tôt ou tard, une responsabilité que chacun de nous devra assumer, en retour de ce qu'ils ont fait pour nous. Face à la perte progressive d'autonomie liée à l'âge, il est nécessaire de réfléchir à la manière d'assurer une qualité de vie digne et sécurisée à ces êtres chers lorsqu'ils deviennent vulnérables. Il devient donc impératif de mettre en place des mécanismes de suivi et d'assistance continue dans des environnements sûrs, en dehors du cadre hospitalier. Ce travail de recherche propose trois contributions principales pour répondre à ces enjeux :

Dans la première, nous concevons un système à base d'agents destiné à la modélisation d'une maison intelligente pour personnes âgées, en utilisant les réseaux de références et l'outil **RENEW** (Reference Net Workshop), en respectant l'architecture **MULAN** (Multi-Agent System Net).

Ensuite, dans la deuxième contribution nous étendons les capacités de **RENEW** en le connectant à un système expert, Java Expert System Shell (**JESS**), afin d'intégrer les aspects délibératifs des agents intelligents. De plus, le modèle proposé tient compte de l'aspect temps pour toute décision.

Enfin, dans la dernière contribution, nous présentons le modèle **ACOSO** (*Agent-Based Cooperating Smart Objects*), une approche fondée sur les systèmes multi-agents (MAS) pour la conception et l'implémentation d'une **pompe à insuline intelligente**, spécialement conçue pour les personnes âgées atteintes de diabète.

Mots clés : Personne âgée, Les réseaux dans les réseaux, Système multi agents mobiles, RENEW, MULAN, JESS, maison intelligente, internet des objets, pompe d'insuline intelligente, ACOSO.

Abstract

Taking care of one's parents becomes, sooner or later, a responsibility that each of us will have to assume, in return for what they have done for us. Faced with the progressive loss of autonomy related to age, it is necessary to think about how to ensure a dignified and secure quality of life for these loved ones when they become vulnerable. It therefore becomes imperative to implement monitoring and continuous assistance mechanisms in safe environments, outside the hospital setting. This research work proposes three main contributions to address these issues:

In the first, we design an agent-based system for the modeling of a smart home for elderly people, using reference networks and the **RENEW** (Reference Net Workshop) tool, respecting the **MULAN** architecture (Multi-Agent System Net).

Then, in the second contribution we extend the capabilities of **RENEW** by connecting it to an expert system, Java Expert System Shell (**JESS**), in order to integrate the deliberative aspects of intelligent agents. Moreover, the proposed model takes into account the time aspect for any decision.

Finally, in the last contribution, we present the Agent-Based Cooperating Smart Objects (**ACOSO**) model, a multi-agent systems (MAS) approach for the design and implementation of an **intelligent insulin pump**, specially designed for elderly people with diabetes.

Keywords: Elderly person, Networks in networks, Mobile multi-agent system, RENEW, MULAN, JESS, smart house, internet of things, intelligent insulin pump, ACOSO.

ملخص

إن رعاية الوالدين تصبح، عاجلاً أم آجلاً، مسؤولية يجب على كل واحد منا أن يتحملها، مقابل ما فعلوه من أجلنا. وفي مواجهة فقدان التدريجي للاستقلالية المرتبطة بالعمر، من الضروري التفكير في كيفية ضمان نوعية حياة كريمة وأمنة لهؤلاء الأبناء عندما يصبحون عرضة للخطر. ومن ثم يصبح من الضروري تنفيذ آليات المراقبة والمساعدة المستمرة في بيئات آمنة، خارج إطار المستشفى.

يقترح هذا العمل البحثي ثلاث مساهمات رئيسية لمعالجة هذه التحديات:

في الأول، قمنا بتصميم نظام قائم على الوكيل لنمذجة منزل ذكي لكبار السن، باستخدام الشبكات المرجعية وأداة **RENEW** (Reference Net Workshop)، مع احترام بنية **MULAN** (Multi-Agent System Net).

بعد ذلك، في المساهمة الثانية، قمنا بتوسيع قدرات **RENEW** من خلال ربطها بنظام خبير **JESS** (Java Expert System Shell)، من أجل دمج الجوانب التداولية للوكلاء الأذكىاء. علاوة على ذلك، فإن النموذج المقترح يأخذ في الاعتبار الجانب الزمني لأي قرار.

وفي المساهمة الأخيرة، نقدم نموذج الكائنات الذكية المتعاونة القائمة على الوكيل (**ACOSO**)، وهو نهج أنظمة متعددة الوكلاء (**MAS**) لتصميم وتنفيذ **مضخة أنسولين ذكية**، مصممة خصيصاً لكبار السن المصابين بداء السكري.

الكلمات المفتاحية: شخص مسن، الشبكات داخل الشبكات، نظام متعدد الوكلاء، المنزل الذكي، إنترنت الأشياء، مضخة

الأنسولين الذكية، **RENEW**، **MULAN**، **ACOSO**، **JESS**.

Table des matières

Résumé.....	4
Abstract.....	5
ملخص.....	6
Introduction générale.....	18

-----État de l'art-----

Chapitre01 :L'Intelligence Ambiante

1.1 Introduction.....	23
1.2 Définition de l'Intelligence Ambiante.....	23
1.2.1 Définition1.....	23
1.2.2 Définition2.....	23
1.3 L'historique de l'intelligence ambiante (AmI)	25
1.4 La vision technique de l'intelligence ambiante.....	26
1.5 Objectif de l'AmI.....	28
1.6 Les caractéristiques de l'AmI.....	28
1.7 Le développement des applications de l'intelligence ambiante	29
1.8 La Sensibilité au contexte.....	29
1.8.1 Définitions et catégorisation du contexte.....	29
1.8.2 Définitions et architecture de la Sensibilité au contexte.....	30
1.8.2.1 Récupération des données contextuelles des capteurs (capture).....	31
1.8.2.2 Traitement des données saisies (interprétation).....	32
1.8.2.3 Identification des situations et du contexte (gestion).....	32
1.8.2.4 Exploitation du contexte.....	32
1.9 L'intelligence Ambiante et les systèmes multi agent.....	32
1.9.1 Les agents ambiants.....	33
1.9.2 Capacités d'un agent ambiant.....	33
1.10 Domaine de l'AmI.....	34
1.11 Les systèmes domotique	35

Table des matières

1.12 La domotique et les personnes âgées.....	36
1.13 Exemples des projets robotique et domotique pour l'assistance à la personne.....	36
1.13.1 Robocare.....	36
1.13.2 IDorm.....	37
1.13.3 MavHome.....	37
1.14 Problématiques propres à la recherche en Intelligence ambiante.....	37
1.15 Les problèmes de l'AmI.....	38
1.15.a Le problème de la miniaturisation.....	38
1.15.b Le problème de l'autonomie énergétique des dispositifs.....	38
1.16 Les avantages de L'AmI.....	38
1.17 Les inconvénients de L'AmI.....	39
1.18 Conclusion.....	39

Chapitre02 : Des réseaux de Petri vers les réseaux de référence

2.1 Introduction	42
2.2 Formalisme de base.....	42
2.2.1 Généralités.....	42
2.2.2 Définition d'un RDP.....	42
2.2.3 Concepts de base.....	43
2.3 Les avantages des réseaux de Petri.....	43
2.4 Quelques extensions des réseaux de Petri de haut niveau.....	44
2.4.1 Les réseaux de Petri colorés.....	44
2.4.2 Les réseaux de Petri temporisés.....	45
2.5 Modélisation des systèmes mobile par les RDPs.....	46
2.5.1 Le paradigme réseaux dans les réseaux.....	46
2.5.2 Le paradigme Nested Net.....	48
2.5.3 Le paradigme réseau objet.....	48
2.5.4 Le paradigme réseau de Petri pour mobilité.....	48
2.6 Réseaux de référence et mobilité.....	49
2.6.1 Réseaux de référence.....	49
2.6.1.a Les instances de réseaux.....	49
2.6.1.b Le réseau est considéré comme Jeton Objet.....	49

Table des matières

2.6.1.c Les canaux synchrones.....	49
2.6.1.d Les types d'arc.....	50
2.6.2 La mobilité.....	50
2.6.2.a Mouvement spontané.....	50
2.6.2.b Mouvement subjectif.....	50
2.6.2.c Mouvement objectif.....	51
2.6.2.d Mouvement consensuel.....	51
2.7 Les systèmes mobile.....	52
2.8 Utilisation des réseaux de Petri dans le domaine des SMAs.....	52
2.9 L'architecture MULAN.....	53
2.10 Outils de modélisation des réseaux de Petri.....	54
2.11 Conclusion.....	54

Chapitre03 : Internet des objets & La méthode ACOSO

3.1 Introduction.....	57
3.2 Internet des objets(IoT).....	57
3.2.1 Définition.....	57
3.2.2 Objet intelligent.....	57
3.2.3 Principe de l'IoT.....	58
3.2.4 Domaines d'application de l'internet des objets.....	59
3.2.5 Les applications de l'IoT.....	59
3.2.6 Internet des objets dans la santé.....	60
3.2.7 Quelques avantages de l'internet des objets.....	61
3.2.8 Les inconvénients de l'intenet des objets.....	62
3.3 Agent Based Computing(ABC).....	62
3.3.1 Principes fondamentaux.....	62
3.3.2 Qu'est ce qu'un agent?.....	63
3.3.3 Système Multi Agent (SMA).....	64
3.3.4 Le paradigme Agent Based computing.....	65
3.3.4.1 L'encapsulation du contrôle.....	65

Table des matières

3.3.4.2 L'interaction.....	66
3.4 Agent based COoperating Smart Objects(ACOSO).....	66
3.4.1 Définition.....	66
3.4.2 L'utilisation du modèle ACOSO dans l'informatique basée sur les agents(ABC).....	67
3.4.3 La méthode ACOSO.....	69
3.4.3.1 Phase d'analyse.....	70
3.4.3.2 Phase de conception.....	70
3.4.3.3 Phase d'implémentation.....	71
3.5 Comparaison entre ACOSO et autres méthodologies.....	72
3.6 Conclusion.....	73

----- Contributions -----

Chapitre04 : Modélisation des maisons intelligentes pour personnes à besoins spécifiques.

4.1 Introduction.....	76
4.2 Maison Intelligente pour les personnes à besoins spécifiques : Spécification.....	76
4.3 Modélisation des composants de base de la maison intelligente.....	77
4.3.1 Modélisation de l'environnement.....	77
4.3.2 Modélisation des agents.....	78
4.3.2.1 Cas d'étude01.....	79
4.3.2.1.a L'agent contexte.....	79
4.3.2.1.b L'agent télévision.....	80
4.3.2.1.c L'agent personne âgée.....	82
4.3.2.2 Cas d'étude 02.....	82
4.3.1.2.a La base de connaissances des agents.....	82
4.3.1.2.b Les protocoles des agents.....	87
4.4 Conclusion.....	89

Chapitre05 : La pompe à insuline intelligente

5.1 Introduction.....92

5.2 La pompe d'insuline.....92

5.3 Spécification, analyse, conception et implémentation du système.....94

 5.3.1 Spécification du système.....94

 5.3.2 Analyse du système.....95

 5.3.3 Conception du système.....98

 5.3.4 Implémentation du système.....102

5.4 Mécanismes de sécurité et protocoles de secours.....103

5.5 Le pseudocode de la smart Pompe à insuline pour le processus décisionnel.....104

5.6 Diagramme de séquence.....106

5.7 La simulation de la pompe d'insuline.....107

5.8 Discussion.....108

5.9 Conclusion.....109

Conclusion générale et perspectives.....112

Références.....115

La liste des Figures

Chapitre01 :

Figure 1.1 : L'Intelligence Ambiante	24
Figure 1.2 : Les domaines technologiques qui composent l'intelligence.....	25
Figure 1.3 : L'historique de l'Intelligence Ambiante	26
Figure 1.4 : Une vision technologique de l'intelligence ambiante.....	27
Figure 1.5 : Un système domotique.....	36

Chapitre02 :

Figure 2.1 : Exemple d'un réseau de Petri simple.....	43
Figure 2.2 : Représentation graphique d'un RDP coloré.....	45
Figure 2.3 : Représentation graphique d'un RDP temporisé.....	46
Figure 2.4 : Un exemple d'un RdP réseaux dans les réseaux.....	47
Figure 2.5 : Un exemple d'un RdP Nets-within-Nets après franchissement.....	48
Figure 2.6 : Mouvement spontané.....	50
Figure 2.7 : Mouvement subjectif.....	51
Figure 2.8 : Mouvement objectif.....	51
Figure 2.9 : Mouvement consensuel.....	51
Figure 2.10 : Système d'agents selon l'architecture MULAN.....	53

Chapitre03 :

Figure 3.1 : Les principes de l'IoT.....	58
Figure 3.2 : Les domaines d'application de l'internet des objets.....	59
Figure 3.3 : Internet of Things et architecture cloud pour la surveillance des soins de santé à distance.....	61
Figure 3.4 : L'architecture à trois couches du modèle ACOSO.....	67

La liste des Figures

Figure 3.5 : La méthodologie ACOSO.....	69
Figure 3.6 : Le méta-modèle de haut niveau utilisé pendant la phase d'analyse de la méthodologie ACOSO.....	70
Figure 3.7 : Le méta-modèle utilisé dans la phase de conception de la méthodologie ACOSO.....	71
Figure 3.8 : Le méta-modèle basé sur JADE utilisé pendant la phase d'implémentation de la méthodologie ACOSO.....	72

Chapitre04:

Figure 4.1 : Exemple d'une maison intelligente.....	77
Figure 4.2 : La maison intelligente comme réseau système.....	78
Figure 4.3 : le réseau principal.....	79
Figure 4.4 : La base de connaissances et les protocoles de l'agent de contexte.....	80
Figure 4.5 : Les protocoles de l'agent TV.....	81
Figure 4.6 : Les protocoles de l'agent personne âgée.....	82
Figure 4.7 : Architecture de JESS.....	83
Figure 4.8 : Le protocole entré au salon.....	87
Figure 4.9 : Le protocole bienvenu.....	88
Figure 4.10 : Le protocole suivi patient.....	88
Figure 4.11 : Exemple de Simulation.....	89

Chapitre05 :

Figure 5.1 : Composant de la pompe d'insuline.....	93
Figure 5.2 : Architecture matérielle de la pompe à insuline intelligente.....	95
Figure 5.3 : Le modèle de SO de haut niveau à la phase d'analyse.....	96

La liste des Figures

Figure 5.4 : Le méta-modèle de la pompe à insuline intelligente de haut niveau pendant la phase de conception.....	99
Figure 5.5 : Organigramme des interactions entre les systèmes de Contrôle.....	100
Figure 5.6 : Le méta-modèle de la pompe à insuline intelligente à la phase d'implémentation.....	103
Figure 5.7 : Le pseudo-code de la smart InsulinPump pour le processus décisionnel.....	106
Figure 5.8 : Diagramme de séquence illustrant le fonctionnement de la pompe à insuline intelligente.....	106
Figure 5.9 : Interface graphique de l'application de la pompe à insuline intelligente.....	107
Figure 5.10 : L'outil Agent Sniffer capture les échanges de messages entre agents dans le système de pompe à insuline intelligente.....	108

La liste des Tableaux

Chapitre 02 :

Tableau 2.1 : Types de mobilité.....52

Chapitre 03:

Tableau 3.1 : Comparaison entre l'ACOSO et certaines méthodologies.....73

Chapitre 04:

Tableau 4.1 : Etats possibles de la télévision.....81

La liste d'abréviations

Abréviation	Signification
ABC	Agent-based Computing
ACL	Agent Communication Language
ACOSO	Agent-based Cooperating Smart Objects
AmI	Ambient Intelligence
AmID	Ambient Intelligence Device
AMS	Agent Management System
BMF	Building Management Framework
CSOs	Cooperating Smart Objects
DF	Directory Facilitator
IDorm	Intelligent Dormitory
IoT	Internet of Things
JADE	Java Agent DEvelopment Framework
MASs	Multi-Agent Systems
MULAN	Multi Agent network
RDP	Réseau de Petri
RFID	Radio Frequency Identification
SMA	Système Multi Agent
Sos	Smart objects
SPINE	Signal Processing In-Node Environment
UML	Unified Modeling Language
WSANs	Wireless Sensor and Actuator Networks
WSNs	Wireless Sensor Networks

Introduction

Générale

Introduction Générale

Dans ce qui suit, nous présentons une vision globale sur les travaux développés au cours de cette thèse, en dévoilant le contexte sur lequel porte notre recherche ainsi que la problématique à résoudre et enfin les contributions apportées par nos travaux.

1. Contexte et problématique

L'intelligence ambiante (AmI) est une notion large qui regroupe un ensemble de technologies visant à nous offrir un espace quotidien « intelligent », facilitant l'accès à l'information ou à des services numériques. Elle permet ainsi une interaction adaptée, naturelle et conviviale avec notre environnement. En ce sens, elle promet un espace technologique capable de comprendre les caractéristiques des usagers et de l'environnement, de s'adapter contextuellement aux besoins, de répondre intelligemment aux demandes ou encore de réagir de manière appropriée à leurs désirs.

Pour atteindre ces objectifs, l'AmI fait souvent appel à un ensemble d'agents sensibles au contexte et dotés d'une certaine intelligence. En effet, ces agents sont capables d'interagir à la fois avec l'environnement réel, avec les usagers, mais aussi de coopérer entre eux afin de résoudre une tâche globale de manière coordonnée.

Dans ce contexte, il est important de noter que le nombre de personnes âgées souffrant d'une autonomie réduite est en constante augmentation. Or, les séjours à l'hôpital ou en maison spécialisée s'avèrent non seulement très coûteux, mais parfois perçus comme inhumains. De surcroît, lorsqu'un incident survient — par exemple une chute — l'intervention des services d'urgence devient complexe et peut exposer la personne concernée à de graves risques.

Par ailleurs, en fonction du type d'incident, différents intervenants peuvent être sollicités : un médecin, une infirmière, un proche, voire un voisin. Face à ces défis, une solution pertinente consiste à maintenir les personnes dépendantes dans leur domicile, tout en assurant un suivi de leur état de santé, de leurs mouvements, ainsi que la détection de tout type d'incident ou de danger potentiel.

Cependant, garder une personne âgée à domicile impose la mise en place d'un environnement adapté, visant à préserver son autonomie tout en assurant une surveillance constante. C'est précisément ici qu'intervient la domotique, un domaine clé d'application de l'intelligence ambiante. Celle-ci propose une solution idéale à travers le concept de maison intelligente (Smart Home).

La conception d'une maison intelligente commence généralement par un modèle de haut niveau, qui est ensuite affiné à travers différentes étapes : simulation, vérification, mise en œuvre et test. Néanmoins, il convient de souligner que la plupart des procédés utilisés dans ce domaine sont encore ad hoc, ce qui entraîne souvent une perte d'informations importantes lors de la transition entre les différentes phases de conception. En conséquence, le système final peut ne pas répondre aux exigences initiales.

En outre, la vie quotidienne dans une maison intelligente impose la prise en compte du facteur temps, qui devient un élément essentiel tant dans la conception que dans la mise en œuvre du système.

Ainsi, l'utilisation d'un modèle formel capable de gérer les particularités temporelles et spatiales propres aux maisons intelligentes permet non seulement de remédier à ces insuffisances, mais aussi de dépasser les limites liées à la nature ad hoc du processus de conception.

2. Contributions

Le formalisme des réseaux dans les réseaux est un paradigme de modélisation formel très puissant qui permet l'étude et la modélisation des systèmes avec des structures hiérarchiques et comportement dynamique tout en tenant en compte les particularités du temps et de l'espace de la maison intelligente. Dans notre première contribution, nous proposons une approche de modélisation et de simulation d'une maison intelligente destinée aux personnes à besoins spécifiques. Cette approche permet d'analyser divers services, notamment ceux liés à la sécurité, au confort et aux aspects économiques. L'objectif principal est de concevoir un environnement domestique capable de soutenir l'autonomie de personnes qui, en raison de besoins particuliers en matière de soins et de santé, seraient habituellement orientées vers des établissements spécialisés.

L'ensemble du système est modélisé selon l'architecture MULAN (Multi Agent Nets), qui est une implémentation du paradigme des réseaux dans les réseaux. Nous mettons particulièrement l'accent sur les agents ambiants cohabitant dans une telle maison, en raison de leurs capacités d'intelligence, d'adaptation, de prédiction et d'auto-organisation. Le modèle obtenu est validé à l'aide de l'outil Renew, qui est un simulateur très puissant et facile à manipuler pour la modélisation des systèmes multi-agents, notamment mobiles. Nous avons également utilisé Jess pour représenter les bases de connaissances des différents agents.

Par ailleurs, le maintien des personnes en bonne santé dans leur propre domicile constitue un enjeu essentiel, notamment pour celles souffrant de maladies chroniques telles que le diabète.

Dans ce contexte, notre deuxième contribution porte sur la manière dont le modèle ACOSO (agent-based cooperating smart objects) améliore la conception et l'efficacité globale des dispositifs de santé dans les maisons intelligentes. Notre approche orientée agent permet une gestion plus efficace des maladies chroniques en assurant une surveillance de la santé en temps réel, une intégration transparente avec d'autres appareils domestiques intelligents, ainsi qu'une intervention plus rapide et pertinente, tout en garantissant la confidentialité et la sécurité des utilisateurs. Utilisant le modèle ACOSO, l'approche proposée, permet une prise de décision plus personnalisée et intelligente par la pompe à insuline intelligente, facilitant ainsi une meilleure gestion de la glycémie chez les patients âgés.

3. Méthodologie

La suite de la thèse est organisée comme suit :

Le premier chapitre nous offrira une compréhension approfondie de l'intelligence ambiante, en abordant son historique, sa définition, ses objectifs et ses caractéristiques. Nous explorerons également les systèmes associés, tels que la domotique, ainsi que le domaine de la sensibilité au contexte, en étudiant la catégorisation du contexte, ses définitions et son architecture.

Le deuxième chapitre met l'accent sur l'approche des réseaux de référence, qui se base sur le paradigme des réseaux dans les réseaux. Dans un premier temps, nous allons présenter les réseaux de Petri avec leurs extensions. Ensuite, nous aborderons quelques paradigmes basés sur les réseaux de Petri, comme le paradigme des réseaux dans les réseaux (Nets-within-nets), particulièrement adapté à la modélisation des systèmes mobiles.

Dans le troisième chapitre, nous présenterons d'abord la technologie relative à l'Internet des objets (IoT), en abordant sa définition, ses principes et ses applications. Par la suite, le paradigme de l'Agent-Based Computing sera examiné en détail, en mettant en évidence ses principes fondamentaux, la notion d'agent, ainsi que la structure et le fonctionnement des systèmes multi-agents. Enfin, le chapitre se conclura par l'étude du middleware ACOSO (Agent-based COoperating Smart Objects), qui illustre de manière concrète la mise en œuvre d'interactions coopératives entre objets intelligents, à travers sa définition et la description détaillée des différentes phases clés de la méthode ACOSO.

Introduction Générale

Le quatrième chapitre sera consacré à la proposition d'une approche de modélisation et de simulation d'un environnement domestique intelligent fondée sur les réseaux de référence. Cette démarche a pour objectif de favoriser le maintien à domicile et l'autonomie des personnes dont la condition nécessiterait, dans un cadre traditionnel, une prise en charge en établissement spécialisé. L'approche proposée s'appuie sur le paradigme des Nets-within-Nets, enrichi par l'intégration du système expert JESS (Java Expert System Shell).

Enfin, le cinquième chapitre présentera une approche basée sur le modèle ACOSO, relevant du paradigme des systèmes multi-agents (MAS), destinée à l'analyse, à la conception et à l'implémentation d'une pompe à insuline intelligente. Ce dispositif s'adresse principalement aux personnes âgées atteintes de diabète et vise à leur fournir des services intégrés de santé, de confort et de sécurité, tout en assurant une surveillance continue et personnalisée de leurs activités quotidiennes.

Finalement, nous dressons une conclusion qui résume nos contributions et présente des perspectives de notre travail.



Chapitre 01

L'Intelligence Ambiante(AmI)

1.1 Introduction

Grâce à la miniaturisation des composants informatiques et des capteurs, ainsi qu'à l'augmentation des capacités de communication, l'informatique connaît des avancées majeures qui ouvrent la voie à l'ère de l'intelligence ambiante. Cette ère repose sur la capacité des utilisateurs à interagir avec de nombreux appareils interconnectés, capables de géo localiser et d'analyser leur environnement. L'intelligence ambiante se présente ainsi comme un moteur d'innovation et de croissance économique, en répondant à des enjeux tels que la réduction de la consommation énergétique, l'amélioration du confort et de l'attractivité des espaces de vie, ou encore l'optimisation de l'efficacité au sein des entreprises.

Ce chapitre vise à fournir une compréhension approfondie de l'intelligence ambiante en abordant son historique, sa définition, ses objectifs ainsi que ses principales caractéristiques. Il présentera également les systèmes qui lui sont associés, notamment la domotique, et s'intéressera au domaine de la sensibilité au contexte à travers l'étude de la catégorisation du contexte, de ses différentes définitions et de son architecture.

1.2 Définitions de l'Intelligence Ambiante

1.2.1 Définition 1

L'intelligence ambiante [63] désigne un concept vaste englobant un ensemble de technologies destinées à un usage partagé. Ce concept repose sur l'idée de petits éléments électroniques, abordables, interconnectés, autonomes, sensibles au contexte et dotés d'un certain degré d'intelligence, intégrés dans notre environnement quotidien (dans nos voitures, nos maisons, dans la rue, etc.).

1.2.2 Définition 2

L'intelligence ambiante [52] a pour objectif de créer un espace quotidien « intelligent » offrant un accès facilité à l'information ou à des services numériques, tout en permettant une utilisation naturelle, adaptée et conviviale. Un tel espace technologique est capable de comprendre les caractéristiques des utilisateurs et de l'environnement, de s'adapter aux besoins contextuels, de répondre de manière intelligente aux demandes ou de réagir de façon appropriée. Cette « intelligence » découle de la convergence des technologies, qui combinent objets intelligents, réseaux de communication et interfaces multimodales pour offrir de nouveaux services aux utilisateurs (figure 1.1).



Figure 1.1 : L'Intelligence Ambiante.

L'intelligence ambiante peut être définie comme l'intersection de quatre domaines technologiques (figure 1.2) :

❖ **L'« Agent ambiant »** : Un système ambiant est constitué de nombreux petits systèmes interconnectés. Ces éléments doivent être compacts, peu coûteux et aussi simples que possible. Ces caractéristiques, spécifiques aux systèmes ambiants, sont principalement traitées au niveau matériel. Plusieurs domaines sont impliqués, tels que l'énergie, l'architecture et le domaine émergent des nanotechnologies.

❖ **La « communication ubiquitaire »** : Elle permet la communication entre les différents éléments et avec les utilisateurs, tout en garantissant une sécurité optimale, notamment en ce qui concerne les données personnelles pouvant être collectées.

❖ **L'« interface utilisateur intelligente »** : Son but est de permettre une interaction intuitive et transparente pour l'utilisateur. Ce domaine inclut des aspects sociologiques, comme les questions éthiques liées à l'analyse et au stockage des données personnelles, ainsi que des aspects techniques relatifs au traitement automatique du langage et à la communication multimodale, qui utilise plusieurs moyens de communication, tels que les gestes, la parole ou les expressions faciales.

❖ **La « sensibilité au contexte »** : Ce domaine repose sur l'idée que les ordinateurs peuvent percevoir et réagir en fonction de leur environnement. Ils doivent être capables de

comprendre la situation dans laquelle ils se trouvent, ainsi que les interactions sociales des objets présents dans leur environnement.

Les applications sensibles au contexte sont des applications dont le comportement s'adapte en fonction du contexte. Un système est considéré comme sensible au contexte s'il utilise ce dernier pour fournir des informations ou des services pertinents à l'utilisateur. Dans les paragraphes suivants, nous aborderons brièvement certaines définitions de la sensibilité au contexte ainsi que les différentes catégories d'informations contextuelles [17].

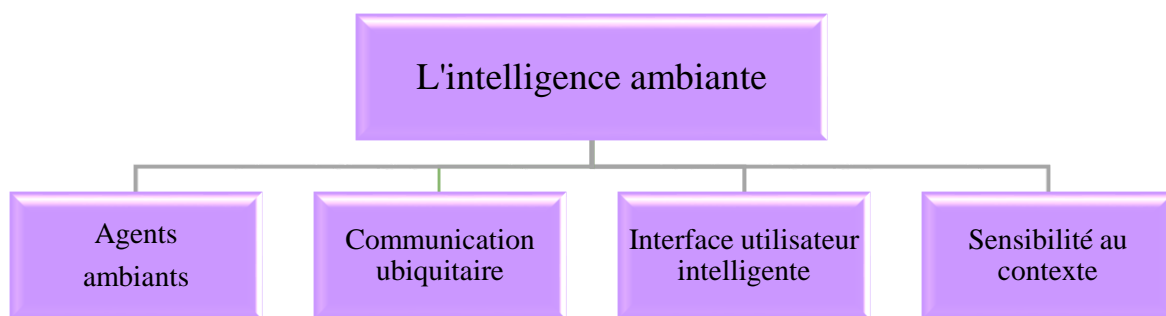


Figure 1.2 : Les domaines technologiques qui composent l'intelligence ambiante.

1.3 L'histoire de l'Intelligence Ambiante (AmI)

L'intelligence ambiante trouve ses racines dans la vision révolutionnaire de Mark Weiser [52], illustrée à la figure 1.3. L'histoire de cette discipline s'inscrit dans une évolution progressive des technologies de l'information et de la communication. Dans les années 1960, l'informatique naît véritablement avec l'apparition des premiers ordinateurs, marquant le début de l'ère du calcul automatique et du traitement électronique de l'information. Durant les années 1970, l'ordinateur commence à communiquer : les réseaux informatiques se développent et posent les bases de l'Internet, ouvrant la voie à l'échange de données entre machines. Dans les années 1990, la mobilité émerge grâce à la miniaturisation des composants et à la diffusion des technologies sans fil, permettant aux dispositifs informatiques de s'intégrer progressivement à la vie quotidienne. Au début des années 2000, les systèmes deviennent ubiquitaires : l'informatique se déploie partout, intégrée dans des objets et des environnements interconnectés, marquant la transition vers un monde où la technologie est omniprésente mais discrète. Enfin, entre 2005 et 2010, l'intelligence devient véritablement ambiante : les systèmes ne se contentent plus d'être présents, ils deviennent sensibles au contexte, capables d'interagir,

d'apprendre et d'anticiper les besoins des utilisateurs, inaugurant ainsi l'ère des environnements intelligents et adaptatifs.

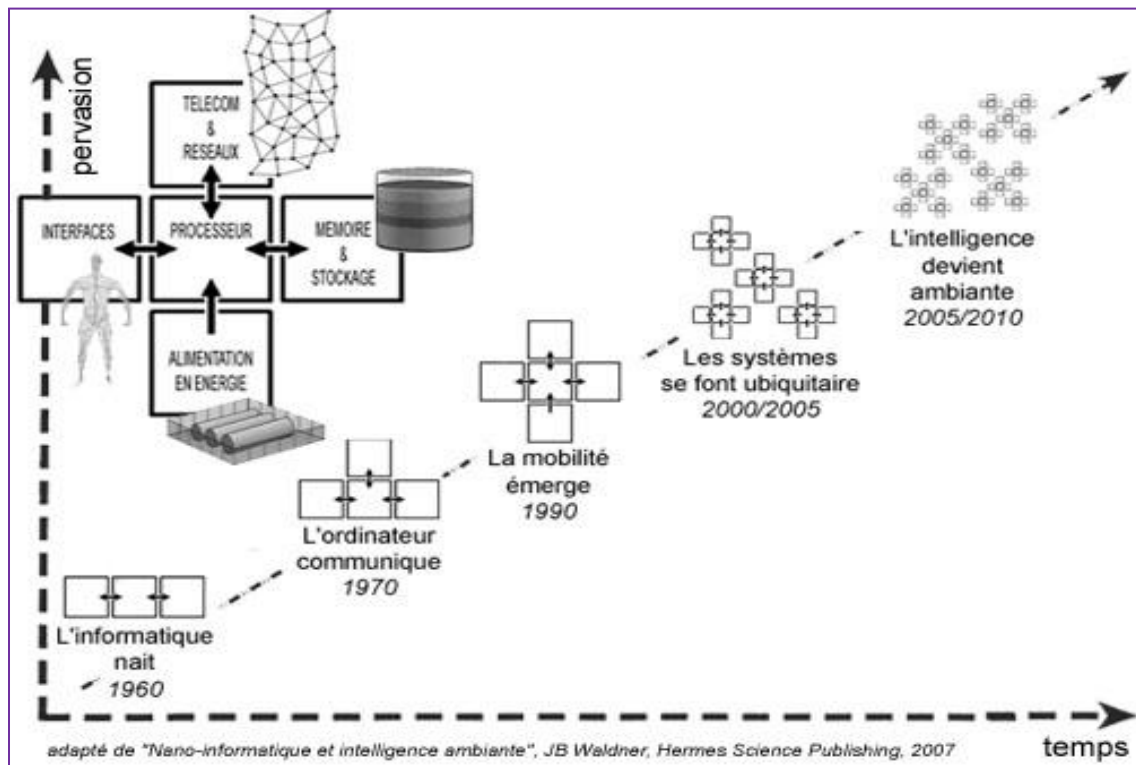


Figure 1.3 : L'historique de l'Intelligence Ambiante.

1.4 La vision technologique de l'intelligence ambiante

La Figure suivante (figure1.4) [13] présente l'ensemble des facteurs d'interaction alimentant l'espace d'intelligence ambiante :

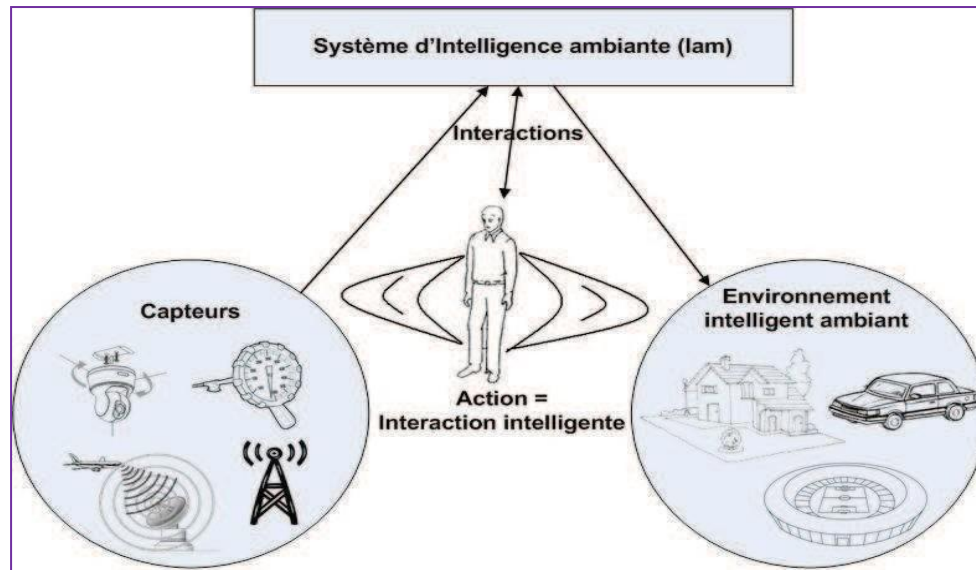


Figure 1.4 : Une vision technologique de l'intelligence ambiante.

Les facteurs d'interaction mis en avant sont les suivants :

✚ **L'environnement** : Il se caractérise par son ubiquité et sa transparence. Il permet de reconnaître les individus, ainsi que certains de leurs besoins ou désirs, et de détecter les changements chez ces individus ou dans l'environnement lui-même. L'environnement peut être varié et multiple, comprenant des lieux tels qu'une maison, une voiture, un hôpital, un stade ou un supermarché. Il peut être modifié automatiquement par le système d'intelligence ambiante ou par l'intervention de l'humain.

✚ **Les capteurs** : Ces dispositifs fournissent au système les informations nécessaires concernant les mesures, la localisation, ainsi que la détection d'objets et de personnes dans l'environnement. Ils sont souvent dissimulés dans l'environnement pour que les interactions de l'utilisateur soient invisibles. En fonction des besoins et du contexte, différents types de capteurs peuvent être utilisés, comme des capteurs de température, de présence, ou de position.

✚ **Le système d'intelligence ambiante** : Il englobe la capacité d'analyse du contexte et l'adaptation dynamique aux situations. Cela permet à l'utilisateur d'interagir, à tout moment et en tout lieu, avec une multitude d'appareils interconnectés autour de lui.

✚ **L'humain** : Il influence l'environnement par une série d'actions, telles que le déplacement, l'interaction ou l'action. Par exemple, dans un hôpital, lorsqu'un patient se déplace d'une pièce à une autre, l'environnement envoie la position et l'identifiant de la personne au système. Selon des critères tels que le type de patient (sexe, maladie, âge) ou d'autres facteurs (température extérieure, taux d'humidité), le système adapte l'environnement

en conséquence, ce qui peut se traduire par des actions telles que le verrouillage de l'accès à certains lieux.

1.5 Objectif de l'AmI

Le domaine de l'Intelligence Ambiante [58] vise un objectif ambitieux : faire collaborer des dispositifs pour soutenir les individus dans l'accomplissement de leurs activités quotidiennes, de leurs tâches et de leurs rituels, de manière simple et naturelle, en utilisant l'information et l'intelligence dissimulées dans le réseau interconnectant ces dispositifs.

Leurs applications seraient nombreuses [3] : allant de la prévention (incendies, accidents) à l'assistance (guidage, contrôle à distance), en passant par le confort. L'une de leurs grandes qualités résiderait dans leur totale transparence : bien qu'ils soient présents, ils seraient entièrement invisibles à nos yeux, et l'interaction avec eux se ferait de manière fluide et discrète.

1.6 Les caractéristiques de l'AmI

Les caractéristiques de l'intelligence ambiante [14] sont :

- **L'ubiquité** : C'est la capacité de l'utilisateur à interagir n'importe où avec une multitude de dispositifs interconnectés, de capteurs, d'activateurs et, plus largement, avec les systèmes électroniques "cachés" autour de lui. Cela se fait via des réseaux adaptés et une architecture informatique hautement distribuée.
- **L'attentivité** : Il s'agit de la capacité du système à détecter en permanence la présence et la localisation des objets, des appareils et des personnes, afin de prendre en compte le contexte d'usage. Cette vigilance est rendue possible grâce à des capteurs répartis dans l'environnement.
- **L'interaction naturelle** : Cela permet à l'utilisateur de communiquer avec le système informatique de la manière la plus intuitive possible, parfois sans nécessiter d'apprentissage préalable, et de façon multimodale, en utilisant alternativement ou de manière synergique plusieurs de ses sens.
- **L'intelligence du système** : C'est la capacité d'analyse du contexte et d'adaptation dynamique aux situations, ainsi que la capacité d'apprentissage, de raisonnement (comme un calcul probabiliste, par exemple), d'inférences et de déductions logiques.

1.7 Le développement des applications de l'intelligence ambiante

Il y a 6Ws questions importantes à répondre quand on veut développer une application de l'AmI [22]:

- ✓ **Qui (Who):** l'identification d'un utilisateur et le rôle que l'utilisateur joue dans 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 peuvent suivre 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 répondre à ces questions, il faut 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.

1.8 La Sensibilité au contexte

1.8.1 Définitions et catégorisation du contexte

La définition du contexte est une étape nécessaire à la construction d'un système sensible au contexte. Pour cela, nous commencerons par citer quelques définitions de contexte qui ont été proposées dans le domaine de l'informatique :

Schilit et al. [68] définissent le contexte comme étant l'information qui répond aux questions « où êtes-vous », « avec qui êtes-vous » et « quelles ressources se trouvent à proximité de chez

vous ». Ils considèrent que le contexte ne se limite pas à la localisation de l'être humain ou à sa situation sociale, mais inclut également des éléments de son environnement tels que : luminosité, niveau de bruit, architecture du réseau et même le coût de la communication. Brown [18] définit le contexte comme une combinaison d'éléments de l'environnement que connaît le système, tels que la localisation, l'heure du jour, la saison de l'année, la température, Etc... Pascoe [59] considère le contexte comme un concept subjectif défini par l'entité qui le perçoit. Pour une entité, le contexte peut être sa localisation spatiale, alors que pour une autre, ce sera sa localisation temporelle ou son état émotionnel. Dey et al. [1] donnent une définition plus générale du contexte. Ils considèrent qu'il s'agit de l'ensemble des informations qui peuvent être utilisées pour caractériser la situation d'une entité. Cette définition est la plus utilisée par les concepteurs de systèmes sensibles au contexte.

Selon les définitions ci-dessus, le contexte peut être défini comme l'ensemble des informations de l'utilisateur (telles que la localisation, la situation sociale et l'état émotionnel) et des informations qui définissent l'environnement (telles que l'heure du système, l'éclairage, la température). On suppose donc que l'information contextuelle peut être classée selon le contexte de l'utilisateur, le contexte physique, le contexte spatio-temporel et le contexte organisationnel :

- ❖ Le contexte de l'utilisateur : Cela représente des informations sur les résidents de la maison et ses préférences telles que : le nom, l'âge, ses maladies, les chaînes de télévision préférées, et le repas préféré.
- ❖ Le contexte physique : comprend des informations sur l'environnement dans lequel se trouve l'utilisateur. Cette catégorie représente le contexte de tous les dispositifs installés dans l'environnement tels que la température et la lumière.
- ❖ Contexte spatio-temporel : Cette catégorie représente des informations sur la position de l'utilisateur et donne une référence temporelle aux activités actuelles de l'utilisateur.
- ❖ Le contexte organisationnel : contient des informations sur la collaboration dans laquelle un utilisateur est engagé, telles que le rôle et l'activité en cours.

1.8.2 Définitions et architecture de la sensibilité au contexte

À mesure que la technologie s'intègre de plus en plus dans notre quotidien, nos environnements deviennent non seulement connectés, mais aussi intelligents. Cette intelligence ne se limite plus à l'exécution d'ordres simples : elle repose de plus en plus sur la capacité des

systèmes à comprendre et à s'adapter à leur environnement. C'est ce qu'on appelle la sensibilité au contexte, une aptitude essentielle pour concevoir des espaces capables d'interagir de manière pertinente avec leurs occupants. Dans ce cadre, la maison intelligente se présente comme un exemple concret de cette évolution technologique.

La maison intelligente peut être considérée comme un système multi-agents [24], c'est-à-dire comme un ensemble d'agents intelligents qui interagissent avec le contexte changeant, où chaque agent représente un appareil installé à la maison. Dans une maison intelligente, diverses interactions ont lieu : entre les appareils, entre les utilisateurs et les appareils qui composent l'environnement physique. Tous les types d'interactions doivent être pris en compte lors de la modélisation et de la simulation des maisons intelligentes. Plusieurs définitions ont été proposées pour la sensibilité au contexte :

Selon Clarizia et al. [23], la sensibilité au contexte décrit la capacité d'un système de percevoir des renseignements sur son environnement, comme l'emplacement, le temps, la température ou l'identité de l'utilisateur, et d'agir en conséquence. Pour Dey et al. [1], une application est sensible au contexte si elle est capable de percevoir son contexte et si elle possède un comportement dynamique guidé par ses connaissances sur son environnement. Une autre définition proposée par [2] établit qu'un système est sensible au contexte s'il peut extraire, interpréter et utiliser des informations contextuelles et adapter sa fonctionnalité au contexte d'utilisation actuel.

Selon les définitions citées ci-dessus, la connaissance du contexte peut être définie comme la capacité de récupérer, d'interpréter, de gérer et d'exploiter l'information contextuelle différente pour mieux adapter sa fonctionnalité. Dans ce qui suit, nous présenterons les différents niveaux qui composent un système sensible au contexte [21, 36, 37, 11] de la récupération des données contextuelles à leur exploitation :

1.8.2.1 Récupération des données contextuelles des capteurs (capture)

Ce niveau permet de récupérer des données contextuelles à partir de capteurs (sensing). Pour cela, plusieurs capteurs sont installés à l'environnement qui interagissent entre eux pour fournir des informations : par exemple, des capteurs de température, des détecteurs de fumée pour détecter les incendies, un capteur de gaz pour détecter le risque de fuite de gaz. Il y a des capteurs d'image et des capteurs de mouvement dans toutes les pièces pour identifier tout visiteur et vérifier s'il est autorisé ou non. Des détecteurs de contact sont également installés sur toutes les fenêtres/portes pour vérifier leur sécurité. De plus, la maison intelligente offre des solutions d'économie d'énergie en installant des capteurs de présence dans toutes les pièces.

1.8.2.2 Traitement des données saisies (interprétation)

Cette phase permet de traiter les données contextuelles fournies par les capteurs. De plus, le fait que les utilisateurs sont au centre des environnements intelligents, le fonctionnement de ces environnements se réfère principalement aux utilisateurs et à leurs activités.

À cette fin, le terme reconnaissance d'activité est souvent utilisé pour l'interprétation des données saisies [7], où les activités des utilisateurs actuels sont modifiées en fonction des données récupérées par les capteurs [74]. Par exemple, le changement d'état d'un capteur fixé à la télévision dans une pièce de vie intelligente à un certain moment est interprété comme l'activité "l'utilisateur regarde un film à l'instant t1" et "l'utilisateur regarde les nouvelles à l'instant t2".

1.8.2.3 Identification des situations et du contexte (gestion)

Dans cette étape, les données contextuelles saisies et interprétées doivent être bien gérées pour faciliter l'utilisation. Ce niveau est utilisé pour représenter les situations contextuelles actuelles. En fonction de ces situations contextuelles, la maison intelligente doit décider quels appareils installés doivent être réagis.

1.8.2.4 Exploitation du contexte

Après avoir identifié des situations contextuelles, il faut exploiter le contexte pour mieux s'adapter. L'adaptation au contexte est l'ensemble des mécanismes de réaction attendus en raison des changements de contexte. Ainsi, une fois que la maison intelligente détecte une nouvelle situation contextuelle, elle doit informer les appareils correspondants pour fournir les services nécessaires. Ces dispositifs communiquent entre eux pour atteindre un objectif commun.

1.9 L'intelligence ambiante et les systèmes multi agents

L'approche Systèmes Multi-Agents (SMA) [2] qui s'est beaucoup développée ces vingt dernières années, permet d'appréhender, de modéliser, de simuler des systèmes complexes. C'est-à-dire les systèmes constitués de nombreux composants en interaction dynamique entre eux et avec le monde extérieur (c'est le cas des systèmes de l'intelligence ambiante). Elle étudie la manière de coordonner un ensemble d'agents afin que ces agents résolvent collectivement un problème global. Ces agents sont autonomes et interagissent via un environnement. L'aspect collectif en SMA est primordial et nécessite l'étude de nouveaux concepts liés à la coordination, la coopération et l'interaction entre les agents.

Nous trouvons que les SMA sont un bon paradigme pour le domaine de l'intelligence ambiante, car des objets interactifs et cognitifs basés sur des agents pourraient répondre aux besoins de l'utilisateur en exploitant des propriétés et des protocoles d'interaction SMA dans des systèmes ouverts de composants en interaction.

Les agents offrent des caractéristiques utiles pour AmI :

- **Réactivité** : Réponds aux évènements ou aux besoins explicites de l'utilisateur.
- **Proactivité** : Agit sans demande explicite pour anticiper les besoins de l'utilisateur.
- **Raisonnement** : Agit de manière intelligente pour prendre la décision correcte dans la situation.
- **Autonomie** : Peut collaborer avec d'autres agents mais peut agir seul, quelle que soit la situation.

Concevoir un environnement d'intelligence ambiante à l'aide des SMA nous emmène à l'utilisation des agents dit « Ambiants ».

1.9.1 Les Agents Ambiants

Un agent ambiant [79] est un agent autonome qui est embarqué sur un dispositif électronique, cet ensemble est appelé *Ambient Intelligence Device (AmID)*. Ces agents sont capables d'interagir avec l'environnement réel, avec d'autres agents et de coopérer entre eux pour résoudre les tâches du système. Tous ces AmID permettent de définir les tâches et les comportements des dispositifs existants dans un environnement AmI. Les AmID offrent des capacités et des comportements qui permettent une bonne interaction entre les dispositifs.

1.9.2 Capacités d'un agent ambiant

Les capacités d'un agent ambiant sont [79]:

- ❖ **Perception de l'environnement** : pour l'agent son environnement est la *vision locale* qu'il a du monde, c'est-à-dire, tout ce qu'il perçoit avec ses capteurs ou les informations reçus des autres agents.
- ❖ **Interaction et collaboration** : l'agent est une unité autonome et en même temps une partie de tout le système, par conséquent, l'agent doit s'intégrer dans le système et effectuer une collaboration avec les autres agents du système pour arriver à réaliser les objectifs du système.

- ❖ **Embarquement sur un dispositif électronique** : l'agent ambiant a la capacité d'être embarqué sur un dispositif électronique, par conséquent, l'agent devient l'entité qui contrôle l'appareil sur lequel il est embarqué.
- ❖ **Survivre au mieux** : l'agent a un cycle de vie (percevoir, décider et agir) sensé être infini, mais le fonctionnement de l'agent dépend de l'énergie de l'appareil qui le supporte. L'agent doit considérer cela dans son cycle de vie de manière à s'adapter aux caractéristiques de l'appareil en ralentissant et/ou suspendant ses activités. De même, il doit déléguer ses fonctions aux autres agents qui ont des capacités supérieures.
- ❖ **Intégrer un nouveau système** : Comme l'agent ambiant est potentiellement mobile, il peut changer d'environnement et par conséquent changer de système.

1.10 Domaines de l'AmI

Elle est appliquée à de nombreux champs de recherche comme [24] : la santé, l'éducation, la sécurité, la vidéo surveillance, la domotique, le transport public... etc. Voici quelques exemples :

- **La domotique et la construction de maisons intelligentes** : La domotique est l'un des domaines de recherche les plus explorés, car il touche directement la vie quotidienne. Dans ces maisons [73], des appareils équipés de capteurs peuvent collecter des informations sur leur utilisation et automatiser diverses tâches. Les principaux avantages attendus sont une amélioration de la sécurité, du confort et des économies d'énergie. Les maisons intelligentes permettent aussi d'offrir un environnement plus sûr, ce qui est particulièrement important pour les personnes ayant des besoins spécifiques, comme les personnes âgées ou malades, qui nécessitent une surveillance constante. Cela réduit les risques en facilitant le suivi des tâches quotidiennes, et permet de détecter des situations anormales pour alerter les services d'urgence en temps réel.
- **La santé et les hôpitaux** : L'Intelligence Ambiante dans le secteur de la santé vise à améliorer l'efficacité des services. Par exemple, des systèmes de surveillance automatisés permettent d'effectuer des analyses en temps réel. Des technologies de sécurité permettent également de restreindre l'accès aux lieux, appareils ou données en fonction des priorités et des autorisations des utilisateurs [3]. Les établissements médicaux deviennent de plus en plus sophistiqués, entraînant une augmentation de la génération et de l'échange d'informations.

Toutefois, il est essentiel de stocker ces informations de manière à garantir à la fois leur efficacité et leur sécurité, tant pour leur gestion que pour leur utilisation.

- **Dans l'éducation** : Les applications d'intelligence ambiante en éducation visent à surveiller les progrès des élèves, leur présence et à mémoriser leurs habitudes pour fournir des conseils sur leur santé [24]. Elles sont également utilisées pour coordonner les emplois du temps, les programmes d'enseignement, etc. Par ailleurs, ces technologies peuvent aider les étudiants à trouver d'autres personnes ayant des intérêts similaires, pour des activités comme le covoiturage. Cet ensemble d'applications est souvent désigné sous le terme de "campus ambiant". En outre, la recherche de personnes partageant les mêmes objectifs ou centres d'intérêt peut également être appliquée à d'autres contextes, comme dans un village ou un centre de loisirs.

- **Dans le secteur des transports routiers** : L'objectif est de rendre le trafic plus fluide et le transport plus efficace et sécurisé. Les technologies comme le GPS, qui permettent de trouver le meilleur itinéraire, aident à éviter les obstacles tels que les travaux ou les accidents, et fournissent des informations sur le trajet (route, durée, etc.). Le paiement électronique sur autoroute est également un exemple d'application. De plus, des techniques de stationnement automatisé des véhicules sont en cours de développement pour améliorer la gestion du stationnement.

1.11 Les systèmes domotiques

La domotique [77] regroupe l'ensemble des techniques et technologies permettant de superviser, d'automatiser, de programmer et de coordonner les tâches de confort, de sécurité, de maintenance et plus généralement de services dans l'habitat individuel ou collectif (Figure 1.5). Le mot est un néologisme, formé à partir du mot latin domus, qui signifie le domicile, et l'automatique que l'on retrouve dans l'électronique, robotique et l'informatique... etc .



Figure 1.5 : Un système domotique.

1.12 La domotique et les personnes âgées

Il est important pour les personnes âgées de garder une certaine autonomie en plus d'une vie sociale, et bien sûr de rester à leur domicile le plus longtemps possible. L'une des études récentes basées sur le concept du « Vieillir chez soi » a révélé que près de 90% des personnes âgées ne résidant pas dans une maison de soins ou d'accueil spécialisée ont affirmés que rester chez eux était très important pour eux. Aujourd'hui, la domotique nous apporte la solution idéale pour préserver l'autonomie des personnes âgées [82] tout en les gardant sous une surveillance permanente grâce aux dispositifs intelligents apportés par la domotique.

1.13 Exemples de projets robotique et domotique pour l'assistance à la personne

1.13.1 RoboCare

Le projet RoboCare [14] initié par l'ISTC (Institut for Cognitive Science and Technology) en Italie vise à rendre indépendance et qualité de vie aux personnes âgées à domicile en intégrant différentes technologies. L'objectif du projet étant de trouver les meilleures façons de combiner les aidants et la technologie intégrant un ou plusieurs robots, afin que l'ensemble agisse dans un cadre de coopération harmonieuse au service de la personne. Le projet combine différentes techniques d'intelligence artificielle et la vision computationnelle. Cette dernière consiste à installer des caméras placées dans l'environnement qui peuvent détecter la présence de l'utilisateur et suivre son activité quotidienne. Les acteurs de ce projet cherchent des solutions basées sur l'intelligence artificielle, notamment les systèmes multi-agents pour offrir

des services personnalisés à la personne à son domicile. Le robot est considéré de façon immergée et donc complètement transparente comme un objet appartenant à l'environnement ambiant.

1.13.2 IDorm (Intelligent Dormitory)

IDorm [14] est un projet développé par le département d'informatique de l'université d'Essex au Royaume-Uni. IDorm est une plateforme composée d'une pièce pour plusieurs usages : dormir, faire des loisirs, travaillé, etc. L'habitat est doté de plusieurs capteurs comme des capteurs de température, de présence, d'humidité mais aussi de plusieurs actionneurs pour ouvrir ou fermer la porte, allumer ou éteindre les radiateurs, ouvrir ou fermer les volets. Pour le contrôle de l'environnement et sa visualisation, l'utilisateur dispose d'une application de réalité virtuelle.

1.13.3 MavHome

MavHome [14] est un projet de recherche multidisciplinaire de l'université du Texas qui est basé sur la construction d'une maison intelligente avec diverses fonctions. Les capacités de la maison sont : le contrôle adaptatif de la température, de l'eau, de la lumière, du nettoyage et du loisir multimédia (audio et vidéo). Il automatise le contrôle de nombreux appareils dans la maison. Pour aborder le problème, les agents doivent assumer le contrôle des tâches plus simples, ces tâches comprennent chaque robot ou chaque capteur.

1.14 Problématiques propres à la recherche en Intelligence Ambiante

L'AmI [3] n'est pas réellement un domaine de recherche en soi. Elle est l'exemple typique de ce qu'est l'ingénierie technologique. Elle se situe à la jonction de plusieurs domaines de recherche, souvent peu habitué à coopérer (Architecture, Interaction Homme Machine, logique). Du fait des nombreux domaines auxquels l'AmI fait appel, seuls des grands organismes sont aptes à mener des recherches sur le sujet. L'AmI étant principalement une technologie visant le grand public, ces organismes doivent comporter en leur sein une ambition commerciale. Ces trois axes (recherche, ingénierie, et commercialisation) font que peu de structures existantes sont à même de comporter un laboratoire spécifiquement dédié à l'AmI. Ainsi, des consortiums se sont mis en place depuis quelques années entre laboratoires et grandes entreprises afin de promouvoir la recherche dans ce domaine. Grâce à ces consortiums, des projets ont pu naître, souvent à échéance lointaine (une dizaine d'année). De ces projets, des nouveaux axes de recherches ont émergés, souvent transversaux. Ces structures ont réussi à

cristalliser les recherches autour de thèmes précis, ce qui est un grand avantage en termes d'organisation et de vitesse de développement.

1.15 Les problèmes de l'AmI

1.15.a Le problème de la miniaturisation

Pour s'intégrer parfaitement dans le quotidien des utilisateurs, [3] les objets devront, non seulement être portables, mais surtout intégré dans les objets d'utilisation. Cette spécification impose une forte miniaturisation des technologies existantes, par exemple en matière d'affichage avec des écrans intégrés aux lunettes, ou en matière de technologies audio.

1.15.b Le problème de l'autonomie énergétique des dispositifs

Le problème de l'autonomie énergétique devient de plus en plus problématique du fait de la miniaturisation des systèmes [3]. Le stockage de l'énergie doit se faire dans un espace très restreint. D'une part les types d'utilisation prévus n'autorisent pas un renouvellement de l'énergie de ces systèmes. La première solution consiste à créer des dispositifs dont l'autonomie énergétique est égale à leur durée de vie (autonomie énergétique longue). D'autre part, il apparaît la nécessité de trouver des algorithmes d'optimisation de consommation énergétique au sein d'un réseau de dispositifs mobiles faisant chacun office de nœuds, du fait du fort besoin énergétiques local que requiert cette fonction.

En effet, selon les opérations à effectuer, la consommation d'énergie varie énormément. L'envoi d'un message est une opération peu coûteuse par rapport à la fonction de relais de communication entre les dispositifs.

1.16 Les avantages de L'AmI

L'intelligence ambiante (AmI) présente de nombreux avantages qui justifient son intégration progressive dans notre quotidien. L'un des apports majeurs réside dans l'économie d'énergie, notamment grâce à l'utilisation croissante des énergies renouvelables comme les panneaux solaires ou les pompes à chaleur, que ces systèmes intelligents peuvent optimiser en fonction des besoins réels. Elle contribue également à améliorer le confort de vie, en permettant un contrôle centralisé et automatisé des équipements domestiques (chauffage, volets roulants, éclairage, etc.), souvent via un ordinateur ou un assistant vocal. L'AmI favorise aussi la mobilité et l'évolution des modes de vie, en déléguant certaines tâches répétitives ou chronophages à des dispositifs intelligents, ce qui permet à l'utilisateur de gagner un temps précieux. Par exemple, dans le domaine de l'automobile, des fonctions telles que l'aide au freinage ou l'allumage automatique des feux illustrent bien cette capacité à assister l'humain dans ses actions

quotidiennes. Un autre avantage majeur de l'AmI est sa capacité à proposer des solutions personnalisées, adaptées aux préférences et aux habitudes de chaque utilisateur, grâce à l'analyse des données collectées. Enfin, ces systèmes intelligents facilitent la communication entre les différents appareils connectés et permettent un partage d'informations fluide, ouvrant la voie à une interopérabilité renforcée entre les technologies et à une meilleure qualité de service.

1.17 Les inconvénients de L'AmI

L'un des principaux inconvénients de l'intelligence ambiante (AmI) réside dans la dépendance croissante qu'elle engendre chez l'être humain, qui risque de devenir esclave de la machine. À force de confier aux technologies la gestion de tâches quotidiennes, voire de décisions personnelles, l'individu peut progressivement perdre son autonomie et sa capacité à agir sans assistance technologique. Par ailleurs, l'AmI repose sur **une** surveillance constante de l'environnement, impliquant la collecte et l'analyse de données en temps réel. Cette présence intrusive soulève des questions éthiques majeures, car l'humain n'est plus libre de ses gestes ni de ses déplacements sans être potentiellement observé ou enregistré. Enfin, ces systèmes peuvent représenter un obstacle à la liberté individuelle, en créant de nouveaux besoins artificiels et en imposant des usages dictés par la technologie elle-même. Cette ingérence dans la vie privée peut aller jusqu'à dicter le comportement des individus, menaçant ainsi leur libre arbitre et leur capacité à rester maîtres de leur propre vie.

1.18 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, et de là, soutenir la sauvegarde de notre planète. Parce qu'elle affectera, par essence, tous les aspects de notre vie quotidienne, l'intelligence ambiante est appelée à devenir un pilier déterminant de l'économie. 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. Le terme intelligence ambiante traduit l'émergence d'un nouveau domaine scientifique. À travers les chapitres suivants nous allons présenter une approche

formelle qui s'appuie sur les réseaux de Petri pour la modélisation d'un modèle d'intelligence ambiante en utilisant la notion d'agent.

Chapitre 02

Des réseaux de Petri vers Les réseaux de référence

2.1 Introduction

Les réseaux de Petri (RdP) constituent un modèle fondamental formel qui offre un outil à la fois graphique et mathématique permettant de modéliser le comportement dynamique des systèmes à événements discrets. Ils sont utilisés dans de nombreux domaines, et sont adaptés à la modélisation de systèmes distribués, ou répartis.

Le présent chapitre met l'accent sur l'approche des réseaux de référence, qui se base sur le paradigme des réseaux dans les réseaux. Dans un premier temps, nous allons présenter les réseaux de Petri avec leurs extensions. Ensuite, nous aborderons quelques paradigmes basés sur les RdP, comme le paradigme des réseaux dans les réseaux (Nets-within-nets), particulièrement adapté à la modélisation des systèmes mobiles.

2.2 Formalisme de Base

2.2.1 Généralités

Le Réseau de Petri (RdP) est un formalisme de représentation mathématique graphique et d'analyse des automatismes. Ils permettent de représenter des systèmes à fonctionnement séquentiel (par étapes).

Les réseaux de Petri ont été créés par Carl Adam Petri, un mathématicien allemand contemporain [8]. Il a introduit ce concept en 1962 dans sa thèse intitulée «Communication avec des automates». Ce formalisme mathématique, à la fois très général et puissant, permet de décrire les relations entre des conditions et des événements, et de modéliser le comportement de systèmes dynamiques à événements discrets. Grâce à sa grande polyvalence, il peut être utilisé pour modéliser aussi bien des protocoles de communication informatique que des systèmes de production. Cependant, l'usage des réseaux de Petri reste principalement orienté vers la recherche plutôt que vers des applications industrielles.

2.2.2 Définition d'un RdP

Les réseaux de Petri fournissent un outil formel ainsi qu'une représentation graphique efficace pour modéliser et analyser des systèmes discrets, en particulier ceux qui sont concurrents et parallèles. Leur aspect graphique facilite la compréhension du système modélisé, rendant l'analyse plus accessible. De plus, leur puissance mathématique permet de simuler des activités dynamiques et concurrentes. L'un des principaux avantages de ces réseaux réside dans leur capacité à analyser les systèmes modélisés, en s'appuyant sur des modèles graphiques et des équations algébriques [78].

Un réseau de Petri est un graphe biparti comprenant deux sortes de nœuds : les places (P) et les transitions (T). Les arcs de ce graphe relient les transitions aux places ou les places aux transitions. Les places contiennent des jetons ou marques (M) qui vont de place en place en franchissant (aussi dit « tirage de transition ») les transitions suivant les règles de franchissement.

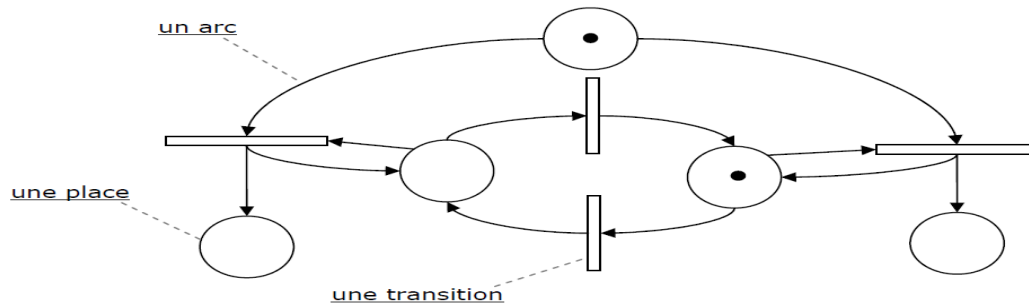


Figure 2.1 : Exemple d'un réseau de Petri simple

2.2.3 Concepts de base

- ✓ **Condition** : Dans un système, une condition représente une proposition logique qui décrit l'état du système et peut être soit vraie, soit fausse. Cette condition est représentée par une place dans le modèle.
- ✓ **Événement** : Les événements représentent les actions qui se produisent dans le système, et leur activation dépend de l'état actuel de celui-ci. Un état spécifique du système peut être considéré comme un ensemble de conditions. Dans le modèle, cette représentation est traduite par une transition.
- ✓ **Déclenchement, pré-condition, post-condition** : Lorsqu'un événement se déclenche, certaines de ses conditions préalables peuvent devenir invalides, tandis que d'autres, appelées post-conditions, deviennent valides. La validité d'une condition est représentée par des jetons dans le modèle.

2.3 Les avantages des Réseaux de Petri

Les réseaux de Petri présentent plusieurs avantages majeurs :

- ✓ Ils offrent une description précise de la structure d'un système, tout en étant simples et accessibles.
- ✓ Ils fournissent un outil de conception graphique, facilitant ainsi la visualisation du système.

- ✓ Ils encouragent une approche itérative de la modélisation, permettant de décomposer progressivement le système en éléments plus simples.
- ✓ Ils permettent d'intégrer à la fois la structure et la dynamique du système sur une seule plateforme.
- ✓ Ils facilitent la transition entre une représentation graphique et une description formelle, ce qui permet une analyse mathématique approfondie assurant la cohérence du système.

2.4 Quelques extensions des réseaux de Petri de haut niveau

La modélisation d'un système réel impose beaucoup de contraintes qui peuvent, mener à des réseaux de Petri de taille importantes. En plus, les RdPs usuels ne permettent pas d'exprimer certaines propriétés, telle que la mobilité, rendant ainsi leur analyse assez difficile. Ceci a motivé plusieurs extensions de réseaux de Petri tels que : les réseaux de Petri coloré [39], ordinaire et temporisé....etc.

2.4.1 Les réseaux de Petri colorés

Lorsque le nombre d'entités à modéliser dans un système est élevé, la taille du réseau de Petri peut devenir rapidement très grande. Dans ce cas, l'utilisation des réseaux de Petri colorés permet de simplifier et de condenser le modèle, surtout si les entités partagent des comportements similaires.

Les réseaux de Petri colorés sont des réseaux où les jetons sont associés à des couleurs. Une couleur représente une information attachée à un jeton, permettant ainsi de les différencier les uns des autres, et cette information peut être de n'importe quel type. Pour accroître l'expressivité d'un réseau de Petri, les jetons placés dans les états du réseau sont colorés, ce qui permet de les distinguer. Ce marquage facilite la différenciation des jetons au sein d'un même état. Un réseau de Petri coloré se caractérise par [39] :

- Chaque place p est caractérisée par un domaine de couleur $C(p)$.
- Un jeton d'une place p est un élément de $C(p)$.
- Chaque transition t est caractérisée par un domaine de couleur $C(t)$.
- Le domaine de couleur d'une transition caractérise la signature de cette transition.
- Les fonctions de couleur sur les arcs déterminent les instances de jetons nécessaires, consommées et produites lors du franchissement d'une transition.

La Figure 2.2 schématise la représentation graphique d'un RDP coloré :

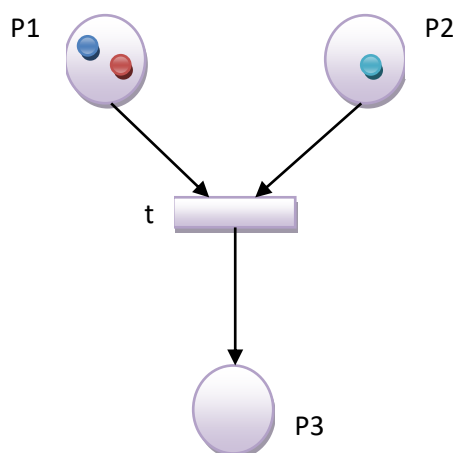


Figure 2.2 : Représentation graphique d'un RDP coloré.

Ainsi, dans un réseau de Petri coloré, les arcs ne sont pas uniquement étiquetés par le nombre de jetons, mais aussi par leurs couleurs. Le franchissement d'une transition dépend alors de la présence, dans les places d'entrée, du nombre requis de jetons, qui doivent également satisfaire aux conditions de couleur des arcs. Après qu'une transition ait été franchie, les jetons correspondant aux couleurs des arcs d'entrée sont retirés des places d'entrée, tandis que ceux correspondant aux arcs de sortie sont ajoutés aux places de sortie de la transition. Bien que les réseaux de Petri colorés ne fournissent pas de puissance descriptive supplémentaire par rapport aux réseaux de Petri classiques, ils permettent une condensation de l'information. Chaque réseau de Petri coloré marqué est isomorphe à un réseau de Petri ordinaire. La relation entre le réseau de Petri coloré et le réseau de Petri traditionnel peut être comparée à celle entre un langage de programmation de haut niveau et le code assembleur. Théoriquement, ces deux niveaux d'abstraction partagent la même sémantique. Cependant, le langage de haut niveau offre une plus grande puissance de modélisation que le langage assembleur, grâce à sa structure, son type et sa modularité [38].

2.4.2 Les réseaux de Petri temporisés

Un réseau de Petri temporisé [76] permet de modéliser un système dont le fonctionnement est influencé par le temps. Les réseaux de Petri temporisés sont particulièrement utiles pour l'évaluation des performances d'un système. La temporisation représente soit la durée minimale de tir, soit le temps de séjour minimum d'un jeton dans une place (ou la durée exacte selon une règle de fonctionnement au plus tôt). Dans ce modèle, la durée d'une activité est explicitement

intégrée. La temporisation peut s'appliquer aux places (réseaux de Petri P-temporisés) ou aux transitions (réseaux de Petri T-temporisés), en fonction des événements modélisés. Dans les réseaux de Petri temporisés [64], chaque jeton reçoit une estampille temporelle qui indique le moment où il devient disponible. Les retards peuvent être utilisés avec des arcs pour contrôler les horodatages des jetons et les durées d'activation des transitions.

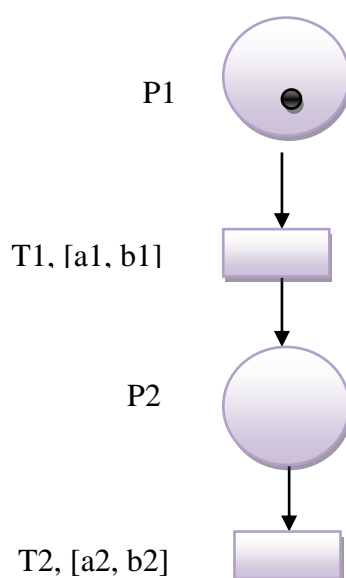


Figure 2.3 : Représentation graphique d'un RDP temporisé.

2.5 Modélisation des systèmes mobiles par les RdPs

La modélisation des systèmes mobiles à l'aide des réseaux de Petri ordinaires ou étendus, tels que les réseaux de Petri colorés, se révèle insuffisante en raison des particularités propres aux applications mobiles (telles que la migration, le clonage, etc.). Par conséquent, il devient nécessaire d'introduire de nouvelles extensions aux réseaux de Petri pour pouvoir modéliser efficacement les applications mobiles.

2.5.1 Le paradigme réseaux dans les réseaux (*nets-within-nets*)

Le paradigme *nets-within-nets* [72] est un formalisme innovant qui permet de modéliser des systèmes en attribuant aux jetons eux-mêmes la structure d'un réseau de Petri. Contrairement aux réseaux de Petri traditionnels où les jetons sont passifs, dans les *nets-within-nets*, les jetons

sont actifs. Ce concept de jeton dans ce paradigme offre une manière simple de modéliser des systèmes récurifs et hiérarchiques. Il permet également de modéliser des systèmes adaptatifs et mobiles.

Dans l'exemple de la Figure 2.4 du réseau de Petri *nets-within-nets*, nous considérons une hiérarchie à deux niveaux. Le niveau inférieur, appelé réseau système, est constitué d'un réseau de Petri dans lequel les places contiennent des jetons qui sont eux-mêmes des réseaux de Petri. Le niveau supérieur, appelé réseau objet, est un réseau de Petri dont les places contiennent des jetons simples.

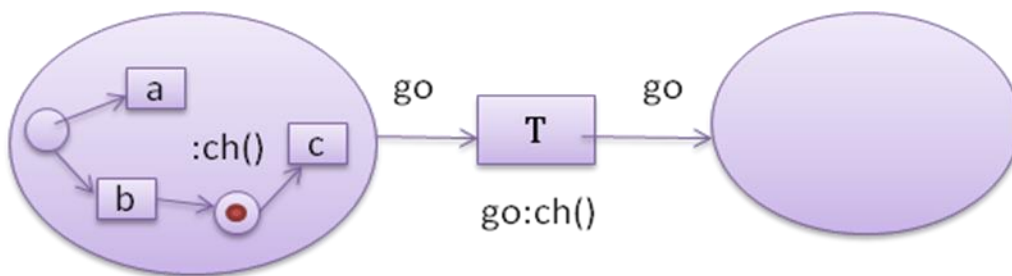


Figure 2.4 : Un exemple d'un RdP réseaux dans les réseaux.

Le réseau objet dans la place de gauche de la figure peut être lié à l'arc avec l'inscription *go*, rendant ainsi la transition *T* active. Remarquons qu'en plus, cette transition est inscrite avec un canal synchrone (*go :ch*). Ceci signifie que pour que le réseau objet *go* soit lié à la transition *T*, une contrepartie doit être vérifiée dans ce réseau objet. Cette pré condition est la transition inscrite par le canal *:ch()* (la transition *c* en l'occurrence). Etant satisfaite dans le réseau objet, on peut dire que le tirage synchrone du réseau objet et du réseau système peut être déclenché. Ceci nous mène à la situation de la figure 2.5 dans laquelle le réseau objet a été déplacé vers la place de droite. En même temps et de façon synchrone, le marquage du réseau objet a changé et désormais, aucun tirage supplémentaire de la transition *ce* n'est plus possible.

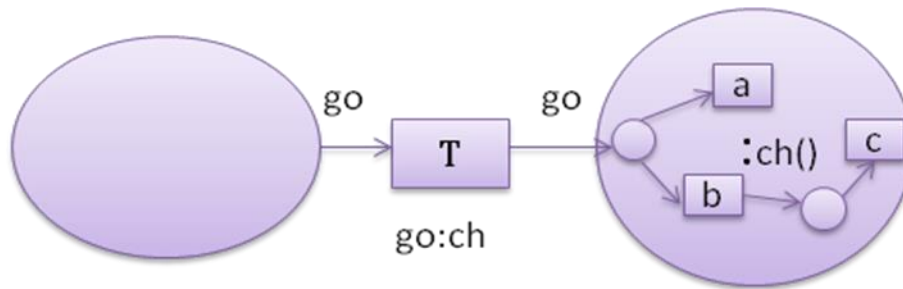


Figure 2.5 : Un exemple d'un RdP Nets-within-Nets après franchissement.

2.5.2 Le paradigme (Nested Nets)

Dans le paradigme *Nested Nets*, chaque jeton peut également être un réseau de Petri. Ce paradigme repose sur les concepts introduits par le paradigme des *réseaux dans les réseaux* (*nets-within-nets*) et s'appuie sur l'étude d'une sémantique différente de la sémantique de référence pour modéliser des systèmes adaptatifs et mobiles [10]. Il constitue une extension des réseaux de Petri colorés, où les jetons peuvent changer de couleur sans nécessiter le franchissement de transitions.

2.5.3 Le paradigme réseau objet (Object Nets for Mobility)

Ce paradigme repose sur le travail des *réseaux dans les réseaux* évoqués précédemment, tout en intégrant des concepts supplémentaires tels que les réseaux de système, les réseaux objet, la fonction de synchronisation, etc. De nouveaux concepts sont introduits dans ce paradigme pour modéliser les systèmes d'agents mobiles. Sémantiquement, trois types de sémantiques peuvent être distingués pour les réseaux objet : la sémantique de référence (Reference Semantics), la sémantique de valeur (Value Semantics) et la sémantique mobile (Mobile Semantics). La sémantique mobile constitue un excellent moyen pour modéliser les applications d'agents mobiles [42].

2.5.4 Le paradigme réseau de Petri pour mobilité (A Petri Net View of Mobility)

Le paradigme réseau de Petri pour mobilité proposé par Charles Lakos [45] est basé sur les concepts suivants :

- Le système contient un nombre des RdPs qu'on appelle modules ou sous réseaux et qui sont interconnectés par des places et transitions de fusion.
- Localisation : est un sous réseau avec une fusion de contexte.
- Sous système : est une localisation non marquée.

2.6 Réseaux de référence et mobilité

Un système mobile peut être décrit comme un ensemble de positions distinctes, dont au moins deux sont obligatoires. Ces positions abritent des entités qui, sous certaines conditions, sont capables de se déplacer d'une position à une autre. Cela implique un changement dans l'environnement de ces entités. Il est important de noter que chaque entité est liée à son environnement local, qui peut l'aider en lui fournissant des services, ou la limiter en lui refusant certains services, soit parce qu'elle n'y a pas accès, soit pour d'autres raisons [43].

2.6.1 Réseaux de référence

Les réseaux de référence sont dits des réseaux de Petri de haut niveau, et sont une notation graphique qui est particulièrement bien adaptée pour la description et l'exécution de tâches complexes et simultanées des processus [69]. Comme pour d'autres formalismes de réseaux il existe des outils pour la simulation des réseaux de référence appelée **Renew (Reference net workshop)** [44]. Les réseaux de référence proposent des extensions pour les réseaux de Petri colorés. On peut citer : les instances de réseaux, les réseaux considérés comme jetons objets, la communication via des canaux synchrones, et différents types d'arcs. A côté de cela, ils sont assez semblables pour les réseaux de Petri colorés tels qu'ils ont été définis par Jensen. Les définitions de ces éléments sont données dans ce qui suit [53], [60]:

2.6.1.a Les instances de réseaux : selon le même principe d'instanciation d'un objet à partir d'une classe dans les langages de programmation objet, les instances de réseaux sont une copie instanciée à partir d'un réseau gabarit (un réseau moule). Notons que différentes instances d'un même réseau peuvent prendre des états différents à un instant donné et sont indépendantes les unes des autres dans tous leurs aspects.

2.6.1.b Le réseau est considéré comme jeton objet : les réseaux de référence implémentent le paradigme des réseaux dans les réseaux. Les places dans ces réseaux (appelés aussi réseaux systèmes) peuvent contenir des jetons schématisant un autre réseau (également appelé réseau objet). Partant de ce principe, on obtient facilement une hiérarchie de réseaux : un réseau système contenant un jeton schématisant un réseau objet peut lui-même être un jeton schématisant un réseau objet dans un autre réseau système.

2.6.1.c Les canaux synchrones : les canaux synchrones ont été proposés pour la première fois par Christensen et Hansen [34]. Ils permettent de synchroniser deux transitions pour les tirer automatiquement en même temps. Pour que cette synchronisation ait lieu, les deux transitions doivent porter le même nom de canal et le même nombre de paramètres. Notons

qu'il est possible de synchroniser plus de deux transitions à un instant donné en les inscrivant, tout simplement, par plusieurs canaux synchrones.

2.6.1.d Les types d'arc : En plus des arcs usuels, les réseaux de référence offrent deux autres types d'arcs : les arcs de réservation et les arcs test. Ces deux types d'arcs sont similaires dans le sens où ils ne changent pas le marquage associé à une place. Les arcs de réservation sont en réalité une autre manière de représenter deux arcs classiques dont les directions sont opposées. Ils permettent de réserver un jeton lors du tirage d'une transition. Les arcs test sont dénués d'orientation et permettent d'accéder (tester) à un jeton donné.

2.6.2 La mobilité

L'outil de modélisation Renew utilise une sorte de pointeur de référence vers (d'où le nom réseau de référence) la fenêtre contenant le réseau jeton. Comme il a été précisé précédemment, dans la terminologie des réseaux de référence, le réseau jeton est appelé *réseau objet* et le réseau le contenant est appelé *réseau système*.

❖ Type de mobilité

L'interaction entre les réseaux objet et les réseaux systèmes permet de capter quatre types possibilités pour qu'un réseau objet se déplace ou se fait déplacer dans un réseau système.

2.6.2.a Mouvement spontané : Le réseau objet est déplacé à l'intérieur du réseau système. Ni le réseau objet, ni le réseau système ne peuvent contrôler le mouvement.

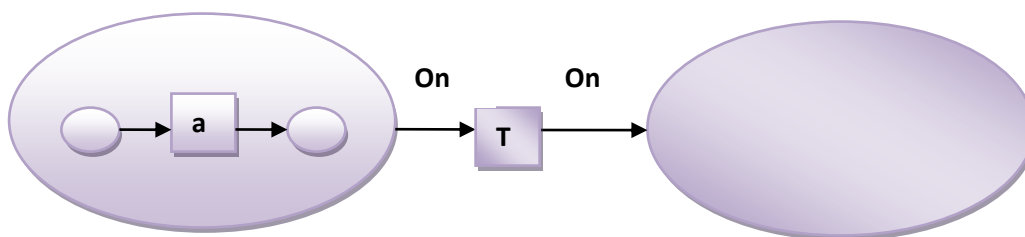


Figure 2.6: Mouvement spontané.

2.6.2.b Mouvement subjectif : Le réseau objet déclenche le mouvement et le réseau système n'a aucune influence sur ce mouvement. Etant donné que le réseau système offre des moyens pour les réseaux objet de se déplacer d'un endroit à un autre, alors le réseau système décide quels sont les mouvements qui peuvent être effectués et qui ne le peuvent pas. Mais dans un réseau système donné, il est possible pour un réseau objet de contrôler le mouvement comme le montre la figure suivante (Figure 2.7). La seule condition pour le mouvement d'être exécuté est le canal synchrone (voir la transition T).

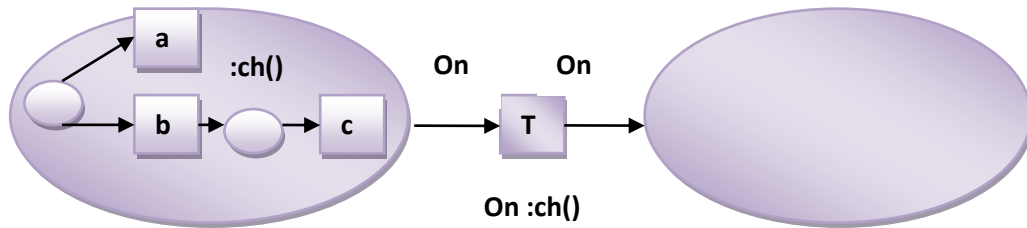


Figure 2.7 : Mouvement subjectif

2.6.2.c Mouvement objectif : Si la figure 2.8 est étendue avec une condition quelconque pour la transition T, le réseau système peut contrôler le mouvement et le réseau objet est transporté d'un endroit à un autre. La figure suivante montre une seule possibilité selon laquelle un réseau système peut contrôler la transition T.

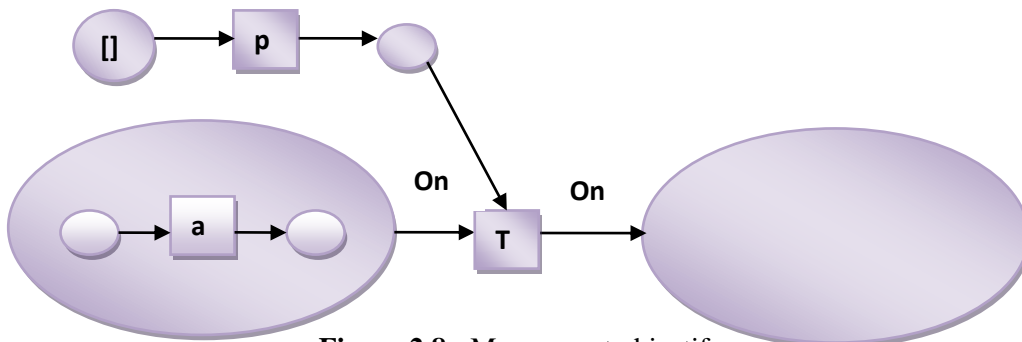


Figure 2.8 : Mouvement objectif

2.6.2.d Mouvement consensuel : En combinant le mouvement objectif et le mouvement subjectif, le réseau objet et le réseau système influencent la transition T.

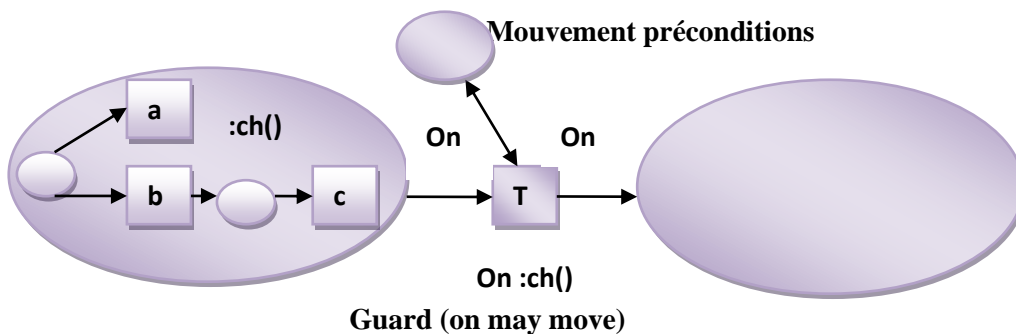


Figure 2.9: Mouvement consensuel.

Les types de mobilité présentés précédemment sont résumés dans le tableau suivant :

Réseau Objet	Réseau système	Types de mobilité	Commentaires
Non impliqué	Non impliqué	Mouvement spontané	Aucune influence du réseau objet et du réseau système sur le mouvement. Aucune pré ou post condition n'est nécessaire.
Non impliqué	impliqué	Mouvement Objectif	Le réseau système contrôle le mouvement, le réseau objet est transporté d'une position à une autre.
Impliqué	Non impliqué	Mouvement Subjectif	Difficile à imaginer dans la pratique, mais théoriquement un réseau objet peut contrôler le mouvement.
Impliqué	impliqué	Mouvement Consensuel	Les deux réseaux ont une influence sur le mouvement

Tableau 2.1 : Types de mobilité

2.7 Les systèmes mobiles

Les systèmes mobiles peuvent être définis comme un ensemble de positions distinctes, dont au moins deux sont obligatoires. Sur ces positions, des entités sont présentes, capables, sous certaines conditions, de se déplacer d'une position à une autre. Cela implique un changement dans l'environnement de ces entités. Il est essentiel de souligner que chaque entité est liée à son environnement local, qui peut l'aider en lui fournissant des services ou la contraindre en lui refusant certains services, soit en raison de leur indisponibilité, soit pour d'autres raisons [41].

2.8 Utilisation des réseaux de Petri dans le domaine des SMA

De nombreux travaux ont été réalisés autour de l'utilisation des RdP pour représenter les modèles multi-agents. Il s'agit, de modéliser l'entité agent moyennant les RdP, les interactions entre agents, l'aspect social (organisation), la migration des agents mobiles,...etc.

Des travaux comme [43] [80] ont été menés sur l'utilisation de RdP pour modéliser les agents mobiles. Les agents sont représentés par des jetons dans des places qui représentent des

positions, les mouvements possibles des agents correspondent au franchissement des transitions.

2.9 L'architecture MULAN

L'architecture multi-agent MULAN (Multi Agent Network) [54] est basée sur le paradigme *nets-within-nets*. Cette architecture est utilisée pour décrire la hiérarchie naturelle d'un système d'agents. L'architecture MULAN facilite la modélisation des agents, surtout si ces agents sont mobiles. La figure 2.10 présente la structure générale de l'architecture MULAN. Cette figure est découpée en quatre parties où chacune décrit un niveau d'abstraction en termes d'un réseau système.

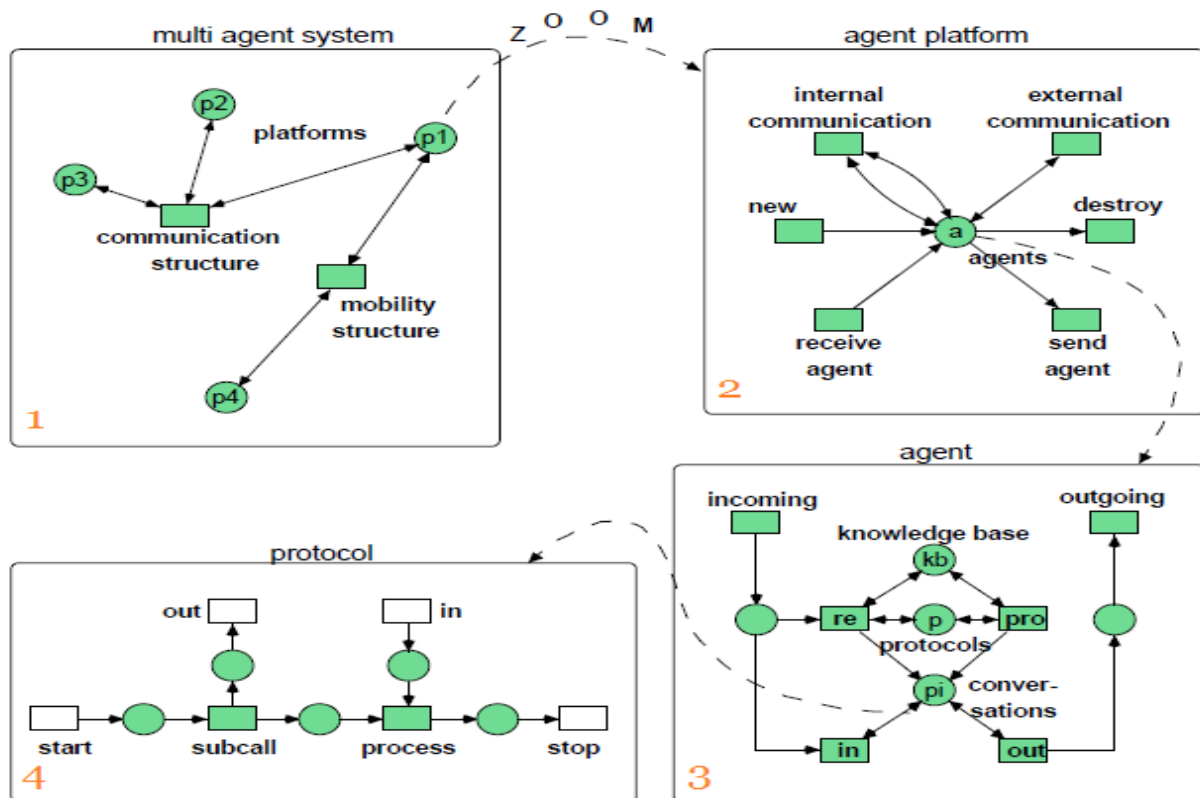


Figure 2.10 : Système d'agents selon l'architecture MULAN

La partie 1 de la figure 2.10 décrit la structure d'un système d'agents où les places contiennent des plateformes comme jeton (chaque plateforme peut contenir un ou plusieurs agents) et les transitions sont considérés comme des canaux de communication ou de mobilité entre les plateformes du système.

En zoomant sur une plateforme de la partie 1 on obtient le réseau représenté dans la partie 2 de la figure 2.10. Cette partie schématise la structure interne d'une plateforme d'un système d'agents. À travers cette structure on distingue qu'une plateforme offre plusieurs services aux agents du système. La transition (*new*) permet la création d'un nouvel agent alors que la transition (*destroy*) permet sa destruction. En ce qui concerne la communication, les deux transitions (*internal communication*) et (*external communication*) permettent l'échange de message entre les agents du système. Deux types de communication sont possibles : une communication interne (entre des agents d'une même plateforme) et communication externe (entre des agents de plateformes différentes). Les deux transitions (*receive agent*) et (*send agent*) permettent le déplacement des agents d'une plateforme vers une autre.

La partie 3 de la figure 2.10 représente la structure d'un agent. On remarque que les agents aussi sont modélisés en termes de réseau. La place (*knowledge base*) contient la base de connaissance de l'agent. Elle définit le niveau d'intelligence de l'agent. Les comportements des agents sont décrits en termes de protocoles (sous forme de réseau également) qui sont placés comme modèles dans la place (*protocols*). La réception d'un message ou un changement dans le contexte autour de l'agent peut déclencher l'instanciation d'un modèle de protocole. Le choix du protocole à instancier est toujours influencé par la base de connaissance de l'agent. Le protocole instancié est placé dans la place (*conversation*). La partie 4 de la figure 2.10 montre un exemple d'un protocole en cours d'exécution.

2.10 Outils de modélisation des réseaux de Petri

Les aspects formels des réseaux de Petri offrent aux développeurs une large gamme d'outils pour simuler et vérifier ces réseaux, en les confrontant aux méthodes de vérification de modèle. Parmi ces outils, on trouve des logiciels tels que CPNTools, CPNAMI, PROD, JARP, MARIA, LOLA, Petri Net Kernel, GreatSPN, Artifex, ExSpect, FLOWer, f-net, GD ToolKit, Helena, HPSim, JFern, JPetriNet, Opera, ORIS, PACE, RENEW, etc. La majorité de ces applications proposent une interface graphique pour la création et l'édition des réseaux de Petri, tout en permettant la simulation de modèles et l'analyse des propriétés générales de ces réseaux.

2.11 Conclusion

Les réseaux de Petri offrent un outil formel et une bonne représentation graphique qui permettent de modéliser et d'analyser les systèmes discrets, particulièrement les systèmes concurrents et parallèles. La facette graphique des réseaux de Petri, nous aide à comprendre

Chapitre02 Des réseaux de Petri vers Les réseaux de référence

aisément le système modélisé. Ce chapitre est consacré aux aspects fondamentaux de la démarche visant une appréhension complète des réseaux de Petri.

Dans le chapitre suivant, nous aborderons le paradigme ACOSO (Agent-based COoperating Smart Objects), en mettant en avant son rôle crucial dans les environnements intelligents et interconnectés, à travers ses trois phases : analyse, conception et implémentation.

Chapitre 03

Internet des objets & La méthode ACOSO

3.1 Introduction

Nous suggérons dans ce chapitre d'utiliser l'approche Agent Based Computing (ABC). Le modèle ABC fait appel à des agents, qui sont des composants logiciels utilisés dans la conception, la programmation, le déploiement et la gestion de systèmes d'Internet des objets (IOT). Le modèle ABC utilise des systèmes multi-agents (MAS) pour la modélisation de systèmes logiciels distribués. Ces agents peuvent effectuer des tâches de façon autonome en communiquant avec d'autres agents et leur environnement par l'intermédiaire du modèle Agent-based COoperating Smart Objects (ACOSO).

Dans cette section, nous présenterons d'abord la technologie de l'internet des objets. Ensuite, nous aborderons le paradigme Agent-based Computing. Enfin, nous présenterons le middleware ACOSO (Agent-based COoperating Smart Objects).

3.2 Internet des objets (IoT)

3.2.1 Définition

L'IoT désigne un concept dans lequel une diversité d'objets, présents en permanence dans l'environnement, peuvent interagir entre eux et collaborer avec d'autres objets. Grâce à des connexions sans fils et des systèmes d'adressage uniques, ces objets sont capables de générer de nouvelles applications et services, tout en poursuivant des objectifs communs [75].

Cette technologie a pour objectif d'interconnecter les individus et les objets, permettant ainsi l'échange et l'automatisation des comportements des objets du quotidien (tels que les capteurs, volets, réfrigérateurs, vidéosurveillance). L'idée est de créer une fusion entre le monde réel (physique) et le monde numérique (virtuel). De ce fait, les objets du monde physique sont intégrés dans l'univers virtuel de l'Internet. Cela engendre de nouvelles tendances et innovations, tant au niveau des architectures de communication que dans la présentation et l'exploitation des services [65].

3.2.2 Objet intelligent

Dans l'ère de l'informatique omniprésente, les objets qui nous entourent seront inévitablement connectés d'une manière ou d'une autre. Grâce aux avancées des technologies de connexion sans fil, ces objets feront partie d'un réseau mondial, l'Internet des objets. Le terme « Objet intelligent » désigne un objet intégré dans ce réseau.

Un objet intelligent peut être défini comme un objet physique enrichi par des capacités technologiques qui lui permettent de communiquer et d'interagir avec son environnement. Le concept d'objet intelligent vise à doter un objet du monde réel de nouvelles fonctions, en complément de celles qui lui sont naturellement attribuées. Ces objets sont ainsi conçus pour récupérer, stocker, transférer et traiter des données de manière continue entre le monde physique et le monde numérique [61].

3.2.3 Principes de l'IoT

L'Internet des objets [66] se compose de plusieurs éléments complémentaires, chacun avec ses propres caractéristiques. Grâce aux systèmes d'identification électronique normalisés et aux dispositifs mobiles sans fil, elle permet d'identifier de manière claire et précise les objets physiques, tout en facilitant la récupération, le stockage, le transfert et le traitement continu des données associées. L'IoT [35] représente un mélange d'innovations technologiques récentes et de solutions préexistantes. Chaque objet dispose d'une identification électronique unique, capable de lire et de transmettre des informations via un protocole sur le réseau internet. Toutefois, il est essentiel de définir la nature de l'objet, ses fonctionnalités, sa localisation, son historique de déplacements, etc (figure 3.1). Pour établir ce lien entre le monde physique et virtuel, le dispositif technique doit modéliser les contextes réels et les reproduire dans le domaine virtuel [66].



Figure 3.1 : Les principes de l'IoT.

3.2.4 Domaines d'applications de l'IoT

L'Internet des objets (IoT) constitue aujourd'hui un pilier essentiel de la transformation numérique, avec des applications diversifiées dans de nombreux secteurs (figure 3.2). Dans le domaine des loisirs, il favorise la personnalisation des expériences et le suivi des performances grâce à des dispositifs connectés intégrés aux équipements sportifs ou multimédias. En matière de transport et de mobilité, l'IoT contribue à l'optimisation des flux de circulation, à la géolocalisation en temps réel et au développement des véhicules intelligents. Les villes intelligentes (smart cities) exploitent ces technologies pour améliorer la gestion des infrastructures urbaines, notamment l'éclairage public, la collecte des déchets et la sécurité. Dans le secteur de la santé, les objets connectés facilitent la télésurveillance médicale et le suivi à distance des patients. L'agriculture connectée tire profit de capteurs environnementaux pour optimiser l'irrigation, la fertilisation et la gestion des ressources. L'industrie, quant à elle, s'appuie sur l'IoT pour renforcer l'automatisation, la maintenance prédictive et la traçabilité des processus. Enfin, la domotique illustre l'application de l'IoT au cadre domestique, permettant un contrôle intelligent et à distance des systèmes de chauffage, d'éclairage et de sécurité.

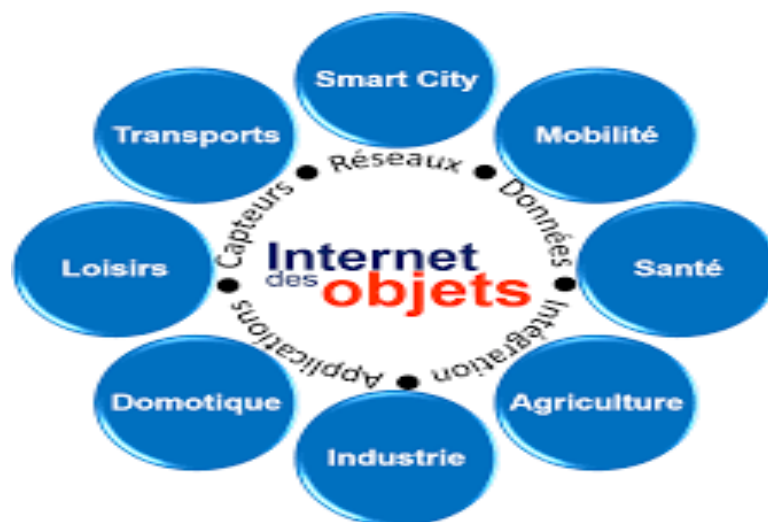


Figure 3.2 : Les domaines d'application de l'internet des objets.

3.2.5 Les applications de l'IoT

L'IoT est utilisé dans divers domaines tels que la sécurité, les systèmes de communication, et la gestion de bâtiments commerciaux et résidentiels, qui intègrent différents systèmes de contrôle pour la gestion du chauffage, de la ventilation, de la sécurité, ou de l'éclairage...etc [70]. Voici quelques exemples d'applications :

- Un réveil qui se déclenche plus tôt que prévu en raison d'un trafic dense. En analysant les données de Google Maps, il ajuste l'heure de réveil en fonction des conditions de circulation.
- Les plantes alertent le système d'irrigation lorsque leur taux d'humidité est trop bas. Les capteurs mesurent des paramètres environnementaux comme la température et l'humidité pour activer l'arrosage.
- Des chaussures de sport fournissent des informations sur le temps, la distance parcourue et les performances de l'utilisateur. Ces données permettent des comparaisons entre athlètes, peu importe leur localisation. Récemment, certaines chaussures sont équipées de puces qui mesurent l'implication de l'athlète et ses performances.
- La gestion des équipements administratifs pour prolonger leur durée de vie, comme les réacteurs d'avion équipés de capteurs qui envoient des flux de données en temps réel. Ces données peuvent atteindre plusieurs To par jour.
- La gestion de l'éclairage public, en réduisant le nombre de lampes allumées ou leur intensité lumineuse lorsqu'aucune présence n'est détectée, grâce à des capteurs de mouvement.
- Le suivi de la santé (Health-care) permet de lutter contre le surpoids ou de détecter des anomalies corporelles (fièvre, hyperglycémie, hypertension, rythme cardiaque trop élevé ou trop faible...) grâce à des capteurs. Ces dispositifs peuvent alerter un professionnel de la santé si un seuil critique est dépassé, et même détecter une chute et signaler si la personne est incapable de se relever [70]. Cet exemple d'application sera détaillé dans ce qui suit.

3.2.6 Internet des objets dans la santé

La santé des individus est de plus en plus reconnue comme essentielle dans le monde actuel en raison des avancées réalisées, bien que de nombreuses pathologies persistent encore. L'augmentation du nombre de patients engendre des délais dans les diagnostics, ce qui rend les patients plus dépendants des médecins pour leurs bilans de santé. Face à ces défis, les systèmes médicaux ont commencé à s'intégrer à l'IoT pour garantir l'identité numérique de chaque patient. Par ailleurs, l'IoT a facilité le suivi constant des progrès des patients et permet aux médecins d'analyser facilement les données collectées [9]. Cette fusion entre la technologie et la santé bénéficie autant aux médecins qu'aux patients. L'Internet des objets devient une technologie de plus en plus prometteuse et évolutive. Ces dernières années, elle s'est révélée particulièrement productive dans le domaine médical, en améliorant la qualité des soins. L'IoT dans le secteur biomédical [47] est désormais plus performant, car la communication entre

médecins et patients a été optimisée grâce aux applications mobiles. Ces applications permettent de suivre en temps réel des informations personnelles du patient (poids, qualité du sommeil, activité physique, etc.) tout au long de la journée ou de la nuit. Elles sont conçues pour permettre aux médecins de surveiller l'état de santé de leurs patients. En cas de problème, le médecin peut intervenir rapidement et fournir le traitement approprié [20]. La Figure 3.3 présente un aperçu général d'un système de surveillance des soins de santé basé sur l'IoT.

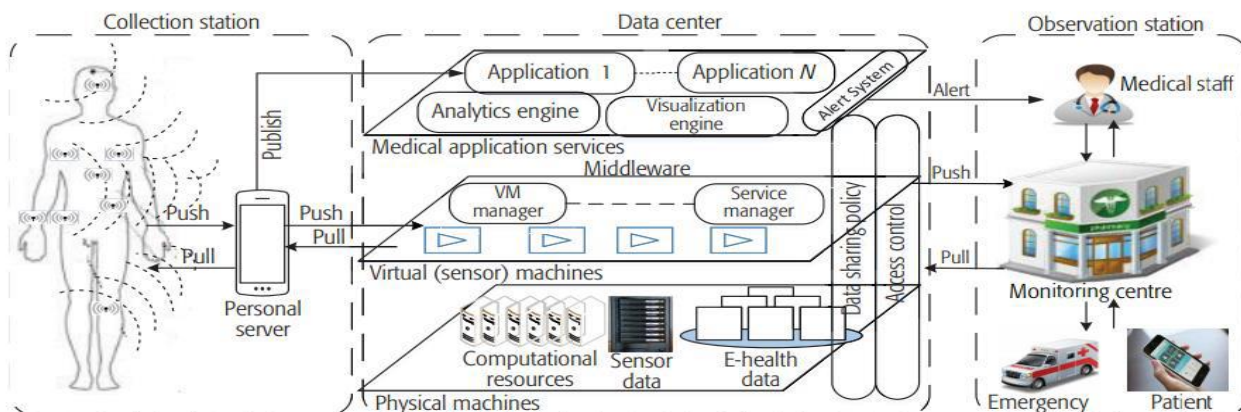


Figure 3.3 : Internet of Things et architecture cloud pour la surveillance des soins de santé à distance [57].

3.2.7 Quelques avantages de l'internet des objets

L'Internet des objets (IoT) présente de nombreux avantages dans la vie quotidienne et dans le monde professionnel. Il permet une meilleure automatisation des tâches grâce à la connexion entre les appareils, ce qui facilite la gestion de l'énergie, la sécurité ou encore la santé. Par exemple, les maisons intelligentes ajustent automatiquement la température ou l'éclairage selon les besoins des occupants. Dans les entreprises, l'IoT améliore la productivité et la maintenance préventive en collectant et en analysant en temps réel des données issues des machines. De plus, il favorise une communication rapide et efficace entre les objets et les utilisateurs, offrant ainsi un gain de temps, une réduction des coûts et une amélioration de la qualité de vie.

3.2.8 Les inconvénients de l'internet des objets

Malgré ses nombreux avantages, l'Internet des objets (IoT) présente aussi plusieurs inconvénients. Le principal risque concerne la sécurité et la confidentialité des données, car les objets connectés collectent et échangent constamment des informations personnelles

susceptibles d'être piratées. De plus, la dépendance excessive à la technologie peut rendre les utilisateurs vulnérables en cas de panne ou de défaillance du système. L'IoT engendre également des coûts élevés d'installation et de maintenance, notamment pour les entreprises ou les foyers souhaitant adopter ces technologies. Enfin, la compatibilité entre différents appareils et le manque de normes universelles compliquent parfois l'intégration et l'utilisation efficace de l'ensemble des objets connectés.

3.3 Agent Based Computing (ABC)

3.3.1 Principes fondamentaux

La première étape pour soutenir une approche orientée agent du génie logiciel consiste à identifier et définir clairement les concepts clés de l'informatique basée sur les agents. La définition et la classification des phénomènes sont toujours des tâches complexes, car il y aura toujours des objections concernant les définitions de base, des arguments sur des aspects importants négligés, ou des affirmations selon lesquelles il ne s'agit finalement de rien de nouveau. Ces critiques sont particulièrement pertinentes lorsque l'entité à définir est intangible et relativement récente. Cependant, ces définitions sont essentielles pour promouvoir le génie logiciel orienté agent. Dans ce contexte, l'approche adoptée ici est de proposer une définition suffisamment large pour englober un large éventail de phénomènes qui peuvent raisonnablement être considérés comme des systèmes basés sur des agents, tout en étant suffisamment précise pour exclure les systèmes qui ne reposent manifestement pas sur des agents [71].

3.3.2 Qu'est-ce qu'un agent ?

Il existe plusieurs définitions du concept d'agent. Voici quelques-unes des plus courantes :

- D'Inverno et Luck [25] définissent un agent comme une entité autonome qui cherche à atteindre les objectifs pour lesquels elle a été conçue.
- Selon Maes [50], un agent est un système informatique situé dans un environnement complexe et dynamique, qui perçoit et réagit de manière autonome afin de réaliser les objectifs pour lesquels il a été créé.
- Selon [27], un agent intelligent est un logiciel qui accomplit un ensemble de tâches prédéfinies, avec un certain degré d'indépendance et d'autonomie, en se basant sur un ensemble de connaissances et une représentation d'objectifs préétablis.

De plus, le concept d'agent [71] peut se décliner sous différentes formes en fonction des besoins des applications. Parmi ces formes, on peut citer :

▪ **Situation** : L'agent est une entité située dans un environnement, capable d'interagir avec celui-ci en fonction des informations sensorielles qu'il en reçoit.

▪ **Autonomie** : L'agent peut agir de manière indépendante, sans intervention extérieure (humaine ou d'un autre agent), et a la capacité de contrôler ses actions ainsi que son état interne.

▪ **Apprentissage** : L'agent est capable d'apprendre et de s'adapter en fonction de ses expériences, modifiant ainsi son comportement en réponse à l'apprentissage accumulé.

▪ **Mobilité** : L'agent peut se déplacer d'une machine à une autre au sein d'un réseau.

▪ **Flexibilité** : Cette caractéristique englobe plusieurs propriétés, notamment la réactivité, la proactivité et la sociabilité.

▪ **Réactivité** : L'agent peut ajuster son comportement en fonction des changements dans son environnement, percevoir celui-ci et fournir une réponse dans un délai requis.

▪ **Proactivité** : L'agent ne réagit pas uniquement à son environnement, mais il agit aussi de manière proactive en fonction d'un objectif, en prenant des initiatives.

▪ **Sociabilité** : L'agent est capable d'interagir avec d'autres agents, qu'ils soient humains ou logiciels, et peut participer à des échanges sociaux ou des transactions.

▪ **Activité** : L'agent est toujours en action, fonctionnant dans un processus ou un thread indépendant.

▪ **Communication** : Cela désigne la capacité de l'agent à échanger des messages avec d'autres agents. Plusieurs schémas de communication existent :

***Now-type messaging** : L'émetteur bloque son exécution jusqu'à ce que le récepteur ait téléchargé le message et envoyé une réponse.

***Future-type messaging** : L'émetteur ne bloque pas son exécution mais conserve une variable pour obtenir ultérieurement le résultat.

***One-way messaging** : Un type de messagerie asynchrone où l'émetteur ne bloque pas son exécution, et le récepteur ne renvoie aucune réponse. Ce type est utile lorsque l'émetteur n'a pas besoin d'une réponse du récepteur pour continuer sa tâche.

3.3.3 Système multi-agent (SMA)

Un système multi-agent (SMA) est constitué d'un ensemble d'agents qui collaborent pour résoudre un problème commun. Ces agents peuvent interagir entre eux de manière indirecte (en agissant sur l'environnement) ou directe (par la communication et la négociation). Ils peuvent choisir de coopérer pour atteindre un objectif commun ou travailler indépendamment pour accomplir leurs propres objectifs [12].

Les SMAs sont généralement conçus pour modéliser des systèmes complexes où de nombreuses entités interagissent entre elles dans un environnement donné. Ces systèmes sont utilisés avec succès dans divers domaines, leurs performances étant évaluées selon les caractéristiques suivantes [26] :

- ❖ **Fiabilité** : elle se réfère à la répartition des tâches entre les agents, facilitant ainsi la résolution rapide et efficace des problèmes au sein du système.
- ❖ **Extensibilité** : elle est liée à l'indépendance des agents les uns par rapport aux autres, permettant ainsi des modifications des comportements des agents sans perturber le système.
- ❖ **Robustesse** : grâce à la coopération entre les agents, le système est capable de tolérer les fautes et de gérer les défaillances éventuelles d'un agent.
- ❖ **Maintenabilité** : grâce à l'interdépendance des agents, chaque agent peut être maintenu séparément sans nuire au bon fonctionnement global du système.
- ❖ **Évolutivité et flexibilité** : l'auto-adaptabilité du système permet aux développeurs d'ajouter ou de retirer des contraintes (voire des agents) sans perturber la structure générale du système.
- ❖ **Efficacité** : la communication entre agents permet de développer des systèmes de calcul distribués puissants et performants.
- ❖ **Réduction des coûts** : contrairement aux systèmes centralisés, qui sont coûteux en termes de développement et de maintenance, les SMA permettent de décentraliser la gestion et l'exécution des tâches, en créant des sous-systèmes d'agents dédiés, réduisant ainsi les coûts d'extension et de maintenance.

3.3.4 Agent-based Computing

L'informatique basée sur les agents (Agent-Based Computing) [48] repose sur le concept d'agent, une entité logicielle avancée qui est autonome, sociale, réactive et proactive. Ces agents évoluent dans un environnement donné, c'est-à-dire dans un monde de ressources perçues, et agissent pour atteindre leurs objectifs, en adoptant des comportements adaptatifs de résolution de problèmes. En interagissant et en coopérant, les agents peuvent résoudre ou accomplir des tâches ou services qui dépassent les capacités d'un seul agent, formant ainsi un Système Multi-Agent (SMA).

En plus de la modélisation, l'ABC constitue également un paradigme de programmation largement reconnu pour implémenter de manière concrète les fonctionnalités avancées des agents et pour répondre efficacement aux exigences des applications modernes, notamment les applications distribuées. En effet, les abstractions des agents, de la société et de l'environnement ont été utilisées pour développer un paradigme de programmation distribuée de haut niveau, fondé sur deux principes clés :

3.3.4.1 L'encapsulation du contrôle : cela consiste à attribuer à chaque agent son propre fil de contrôle et ses capacités de raisonnement, permettant ainsi de concevoir des entités qui prennent en compte leur contexte et présentent des comportements autonomes.

3.3.4.2 L'interaction : cela inclut les mécanismes de coordination et de coopération, fondés sur l'échange de messages asynchrones de haut niveau. Dans ce cadre, l'adoption de normes de communication et de spécifications de gestion communes (telles que les plateformes systèmes et les langages de communication basés sur le FIPA de l'IEEE) permet aux agents de faciliter l'interopérabilité, en intégrant une variété de ressources et de systèmes existants dans la société des agents. Ces avantages contribuent à améliorer les performances du système (notamment l'efficacité du calcul, la fiabilité, la réactivité, etc.), ainsi que l'interopérabilité et l'évolutivité, particulièrement en comparaison avec les approches centralisées [31].

3.4 Agent-based COoperating Smart Objects (ACOSO)

3.4.1 Définition

ACOSO (Agent-based COoperating Smart Objects) est un paradigme de programmation et de conception orienté vers les agents, convient à l'analyse, à la conception et à l'implémentation des objets intelligents, aussi appelés "smart objects", qu'ils interagissent de manière autonome grâce à des agents, dans le but de collaborer pour réaliser des tâches

spécifiques et optimiser l'efficacité et l'intelligence globale du système. Ces objets, généralement dotés de capteurs, d'actionneurs et de dispositifs de communication, sont capables d'agir de façon indépendante tout en travaillant ensemble avec d'autres objets au sein d'un environnement dynamique et interconnecté. L'objectif du middleware est de fournir les bons instruments et un modèle de programmation simple pour réaliser des objets intelligents coopérants.

L'architecture proposée pour les SO est à la fois simple et efficace. Elle se compose d'un comportement qui intègre celui de l'objet, d'un répartiteur d'événements qui gère les événements internes de l'objet, d'un sous-système de gestion des communications qui contrôle les échanges avec d'autres SO et des entités externes, d'un sous-système de gestion des dispositifs qui coordonne les nœuds de capteurs et d'actionneurs de l'objet, ainsi que d'un sous-système de gestion des connaissances qui supervise la base de connaissances du SO. ACOSO est entièrement intégré dans deux réseaux de capteurs sans fil à contrôle de trame, tels que le cadre de gestion des bâtiments (Building Management Framework) et le cadre Environnement de traitement du signal dans le nœud (SPINE) [47].

3.4.2 L'utilisation du modèle ACOSO dans l'informatique basée sur les agents (ABC)

Ces dernières années, le paradigme ABC a été largement adopté dans la recherche et l'industrie dans divers domaines, démontrant son efficacité dans l'analyse, la modélisation et l'implémentation de systèmes distribués complexes, coopératifs et adaptatifs [26]. En outre, le paradigme ABC peut intégrer des agents avec des technologies sémantiques [32], facilitant ainsi la gestion des données et du contexte tout en renforçant les mécanismes de sécurité. L'utilisation d'agents dans de tels systèmes permet d'apporter une connaissance du contexte, de l'intelligence, de la flexibilité et de la robustesse à la fois aux systèmes d'exploitation individuels et à l'ensemble du réseau Internet des objets. Le paradigme ABC permet également la conceptualisation, la simulation et la réalisation de systèmes distribués avec plusieurs entités. Pour tirer parti de ces avantages, le modèle ACOSO peut être utilisé. ACOSO est un middleware basé sur des systèmes d'information et des systèmes événementiels qui utilise la coopération entre agents pour atteindre des objectifs spécifiques. Il s'agit d'un outil précieux pour le développement et la gestion des systèmes d'exploitation, composé de trois couches au sein de la plateforme, comme indiqué ci-dessous (figure 3.4) :

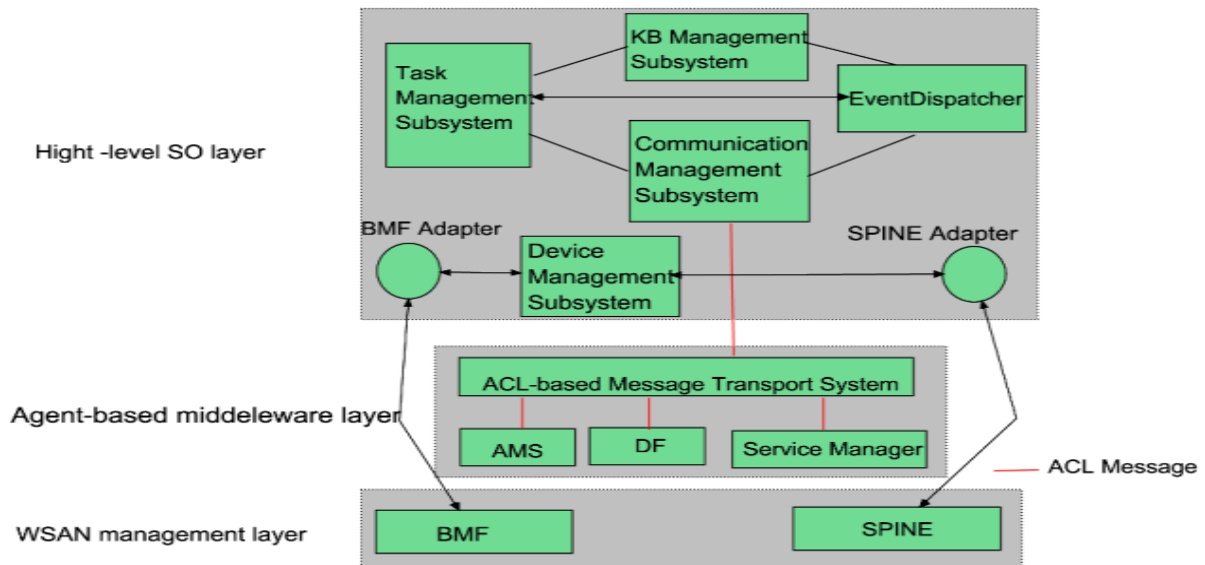


Figure 3.4 : L'architecture à trois couches du modèle ACOSO.

- **Couche de gestion des réseaux de capteurs et d'actionneurs sans fil (WSAN management layer):** Cette couche est responsable de la gestion du réseau de capteurs et/ou d'actionneurs intégrés à un système, ce qui permet le contrôle efficace des réseaux de capteurs et d'actionneurs sans fil (WSAN) au moyen du cadre de gestion du bâtiment (BMF) [29]. De plus, cette couche facilite la gestion des réseaux de capteurs corporels par l'intermédiaire du système SPINE (Signal Processing In-Node Environment) [30].

- **Couche de middleware basée sur l'agent (Agent-based middleleware layer):** Cette couche est construite sur la plate-forme Java Agent DEvelopment Framework (JADE), qui utilise le système de gestion des agents (AMS) pour gérer les OS basées sur JADE. Il permet la communication entre les OS par l'intermédiaire du système de transport des messages basé sur le langage de communication des agents (ACL). Pour rechercher les OS et autres agents, il utilise une version étendue du Directory Facilitator (DF).

- **Couche SO de haut niveau (Hight level SO layer) :** L'architecture interne d'un SO est décrite par une collection de sous-systèmes, chacun contribuant à la réalisation des objectifs du SO. Les activités de l'État correspondant aux objectifs du SO sont exécutées et gérées par le sous-système de gestion des tâches en réponse à des événements. Le sous-système de gestion des communications veille à ce que l'interface de communication du SO demeure uniforme, le gestionnaire de messages du gestionnaire des communications étant responsable de la conversion des messages entrants en événements internes gérés par le dépanneur d'événements [33]. Le sous-système de gestion des appareils contrôle les dispositifs de capteur/actionneur du SO à l'aide d'adaptateurs de périphériques personnalisés. Le sous-système de gestion de la base

de connaissances gère la base de connaissances des objets. Ces sous-systèmes reposent sur des composants logiciels enfichables appelés adaptateurs pour faciliter la communication avec les systèmes externes ou entités.

ACOSO-Meth utilise des méta-modèles positionnés à différents niveaux d'abstraction, totalement indépendants de tout contexte d'application particulier, afin de soutenir méthodiquement le processus de développement de l'IoT. Cette décision confère un certain degré d'universalité à la méthodologie, ce qui est un mérite remarquable compte tenu de la grande variété des scénarios IoT en constante évolution. Indépendamment de la portée, des objectifs ou de l'environnement d'applications OS particulières, les méta-modèles proposés sont neutres en termes de domaine et peuvent être utilisés efficacement. Un modèle de haut niveau décrivant les caractéristiques fondamentales des OS permet à ACOSO-Meth d'appuyer le processus analytique. En raison de sa spécialisation et de son niveau de détail, ce modèle évolue au fur et à mesure qu'il est conçu et mis en œuvre. Plus précisément, la phase d'analyse utilise un méta-modèle de SO de haut niveau. Au cours de la phase de conception, les composantes fonctionnelles du système, leurs relations et leurs interactions sont modélisées à l'aide d'un méta-modèle de niveau analytique basé sur ACOSO.

Un méta-modèle JACOSO (ACOSO basé sur JADE) adapte le méta-modèle SO basé sur ACOSO pour une implémentation spécifique sur la plateforme JADE pendant la phase d'implémentation.

3.4.3 La méthode ACOSO

Pour faciliter les étapes d'analyse, de conception et d'implémentation des SO, le paradigme ABC et l'intergiciel ACOSO sont intégrés, ce qui donne la méthode ACOSO, comme illustré dans la figure 3.5.

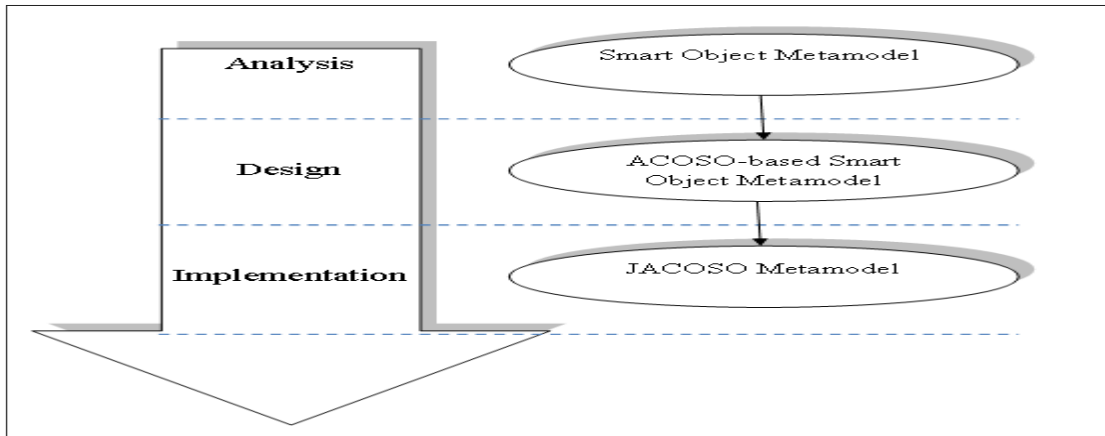


Figure 3.5 : La méthodologie ACOSO.

La méthode ACOSO a pour objectif de soutenir le développement d'objets intelligents (SOs) à travers des méta-modèles qui fonctionnent à différents niveaux d'abstraction et sont totalement indépendants du contexte spécifique de l'application. La méthode ACOSO utilise un modèle de haut niveau pendant la phase d'analyse pour décrire les caractéristiques fondamentales du SO. Ce modèle est ensuite affiné et développé dans les phases de conception et de mise en œuvre. Plus précisément, pendant la phase d'analyse, un méta-modèle de SO de haut niveau est utilisé; la phase de conception utilise un méta-modèle de SO basé sur ACOSO pour définir les composants fonctionnels, leurs relations et interactions; et la phase d'implémentation emploie un méta-modèle de JACOSO pour une mise en œuvre sur la plateforme JADE, comme indiqué dans la recherche par [19]. Les phases clés de la méthode ACOSO sont résumées ci-dessous.

3.4.3.1 Phase d'analyse

Pour fournir un soutien complet pendant la phase d'analyse du développement de SO, la méthode ACOSO utilise un méta-modèle intelligent à objet de haut niveau. Ce méta-modèle décrit les principales caractéristiques, qui sont présentées en utilisant un diagramme de classe du langage de modélisation unifié (UML) (figure 3.6). Ces fonctionnalités décrivent les caractéristiques statiques et dynamiques des OS, ce qui rend le modèle adaptable à divers domaines d'application, tels que les véhicules intelligents, les bâtiments intelligents, etc.

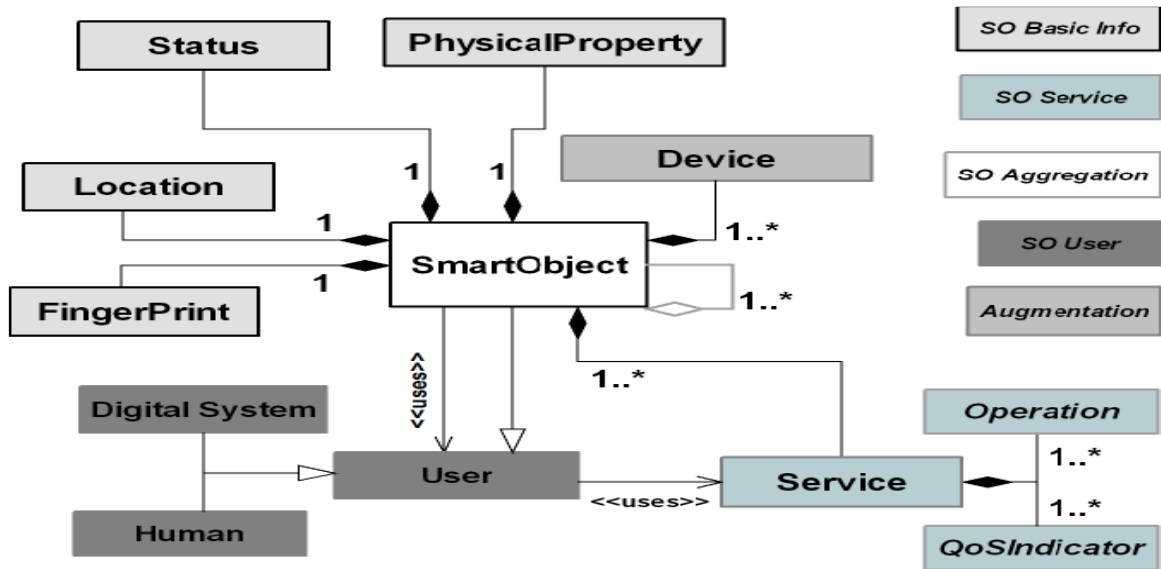


Figure 3.6 : Le méta-modèle de haut niveau utilisé pendant la phase d'analyse de la méthodologie ACOSO.

3.4.3.2 Phase de conception

Afin de faciliter la modélisation des composants fonctionnels, des liens et des interactions au sein d'un système IoT, le méta-modèle SO de haut niveau est développé pendant la phase de conception pour fournir un méta-modèle SO basé sur ACOSO (figure 3.7). Ce méta-modèle SO, basé sur ACOSO, est bien adapté pour la modélisation des composants IoT plus simples tels que les capteurs, les actionneurs et les OS, ainsi que des systèmes plus complexes comme les réseaux de capteurs sans fil (WSN) et les systèmes d'identification par radiofréquence (RFID). Partie essentielle du logiciel intermédiaire ACOSO [28], qui sert à créer, administrer et mettre en œuvre des objets intelligents coopératifs (CSOs) orientés vers les agents, ce méta modèle permet aux développeurs de créer des CSOs qui gèrent efficacement la gestion des connaissances, l'informatique distribuée, proactivité et communication entre les capteurs et les actionneurs dans n'importe quel environnement d'application de l'IoT [4]. Ainsi, il répond à la fois aux besoins du système et aux besoins de haut niveau identifiés précédemment.

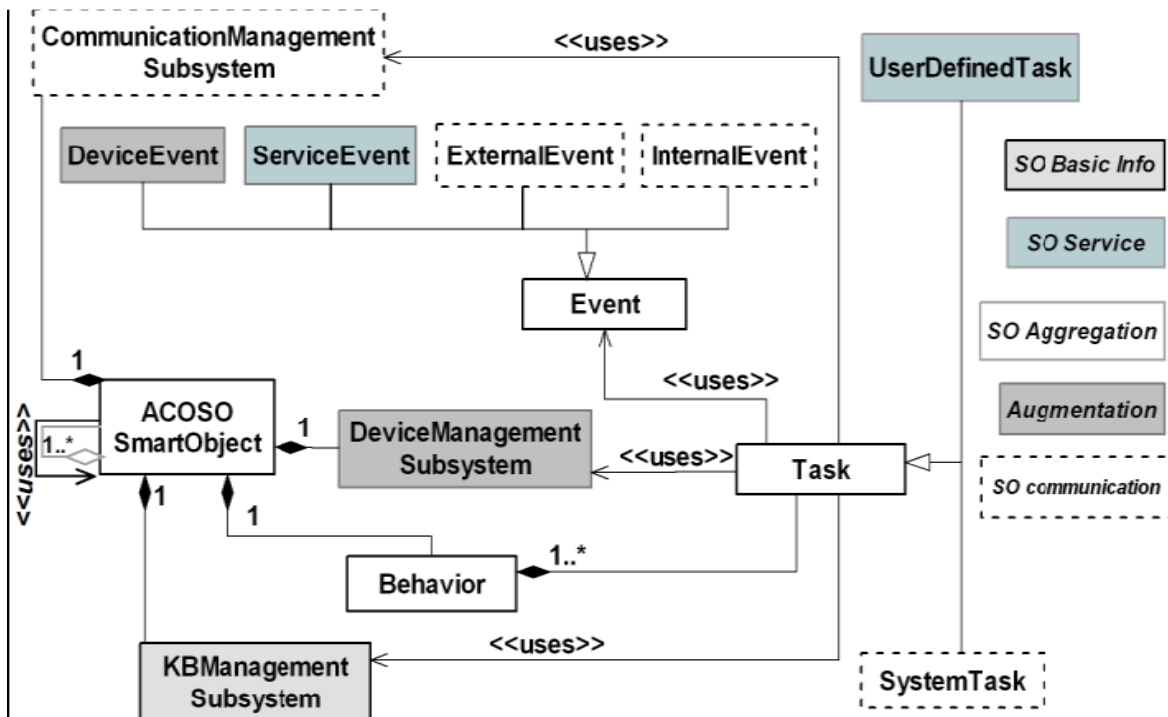


Figure 3.7 : Le méta-modèle utilisé dans la phase de conception de la méthodologie ACOSO.

3.4.3.3 Phase de d’implémentation :

La plateforme JADE est utilisée pour mettre en œuvre le méta-modèle d’SO basé sur ACOSO, ce qui facilite le processus de mise en œuvre [16]. Pour faciliter la découverte des services d’agent, le DF JADE a été amélioré avec une interface axée sur l’agent [12], ce qui permet aux SO de fournisseurs de services d’enregistrer, d’indexer et de rechercher en fonction d’attributs fonctionnels ou non fonctionnels spécifiques.

Le méta-modèle présenté à la figure 3.8 est l’application du méta-modèle SO basé sur ACOSO, appelé JACOSO.

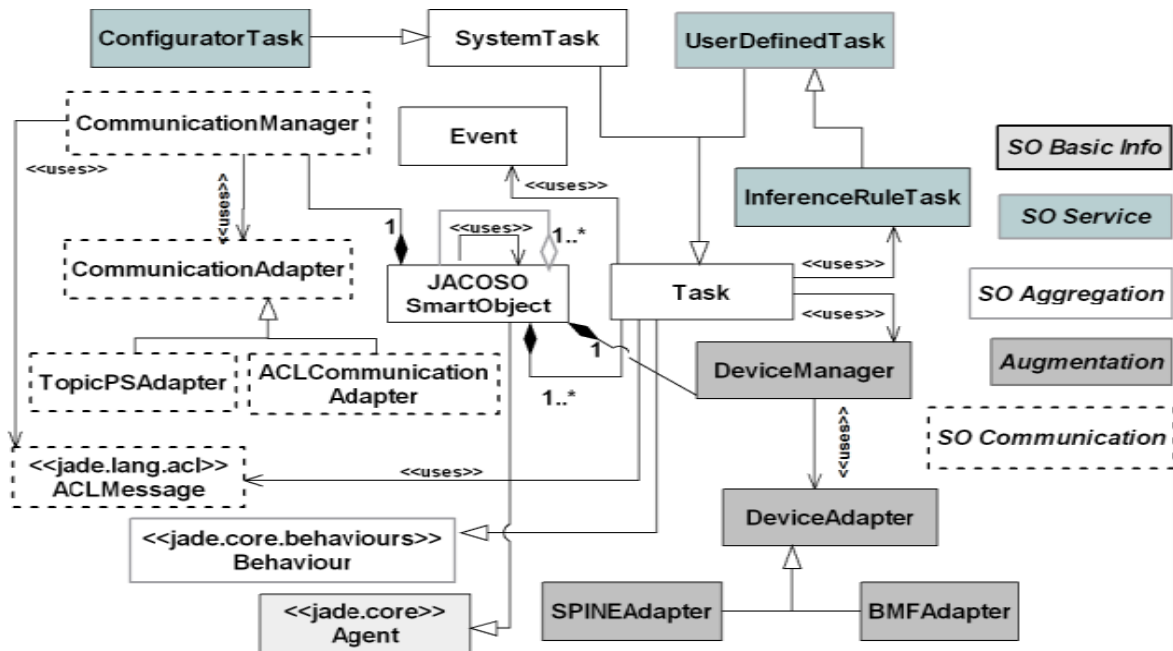


Figure 3.8 : Le méta-modèle basé sur JADE utilisé pendant la phase d'implémentation de la méthodologie ACOSO.

3.5 Comparaison entre ACOSO et autres méthodologies

Dans cette section, nous présentons une comparaison entre ACOSO et quelques méthodologies utilisées pour les dispositifs de santé intelligents. Cette comparaison évalue divers aspects, notamment le type d'architecture, l'adaptabilité, les capacités de prise de décision en temps réel, l'évolutivité, la gestion des pannes et la fiabilité globale. Les résultats sont résumés dans le tableau 3.1.

Caractéristique / Méthodologie	ACOSO (Agent-based COoperatif Smart Object)	Systèmes basés sur des règles [46] (RBS)	Systèmes basés sur le cloud [49]	Edge Computing [62]
Type d'architecture	Basé sur des agents autonomes et coopératifs	Basé sur des règles préprogrammées (if-then)	Centralisé (traitement des données dans le cloud)	Local (traitement des données au plus près des dispositifs)
Adaptabilité	Très élevée (agents s'adaptent à l'environnement)	Faible (dépend des règles préétablies)	Moyenne (dépend du cloud et de la connectivité)	Moyenne (dépend des ressources locales)
Prise de décision en temps réel	Très élevée (réaction rapide grâce à la coopération des agents)	Faible (dépend des règles statiques)	Moyenne (retard possible selon la bande passante et le traitement distant)	Haute (réponses rapides grâce au traitement local)
Scalabilité	Très élevée (facilement extensible avec plus d'agents)	Faible (complexité croissante avec l'ajout de règles)	Très élevée (infrastructure cloud extensible)	Moyenne (limité par la capacité des dispositifs locaux)
Gestion des pannes et fiabilité	Très haute (les agents coopèrent et peuvent remplacer d'autres agents en cas de panne)	Moyenne (pas de redondance prévue, dépend du respect des règles)	Faible (dépend du cloud et de la connectivité)	Haute (les dispositifs peuvent fonctionner indépendamment du cloud)

Tableau 3.1 : Comparaison entre l'ACOSO et certaines méthodologies.

3.6 Conclusion

La révolution systémique induite par la réalisation complète de la vision de l'IoT n'en est qu'à ses débuts. Bien que de nouveaux dispositifs et systèmes IoT aient déjà été développés, nous avons mis en évidence l'importance de paradigmes tels que l'ABC (Agent-based Computing), l'informatique basée sur les agents, les systèmes multi-agents, ainsi que les différentes couches du middleware ACOSO. Une telle approche de développement est essentielle pour soutenir l'intégration de l'IoT. Actuellement, ACOSO est en pleine réorganisation, en exploitant des infrastructures de pointe et des solutions basées sur le cloud, afin de permettre la virtualisation des objets intelligents.

Chapitre 04

*Modélisation des maisons intelligentes pour
personnes à besoins spécifiques.*

4.1 Introduction

La maison intelligente est un domaine d'application qui a attiré beaucoup d'attention dans la communauté de l'intelligence ambiante en raison des applications possibles dans les soins à domicile des personnes âgées. Dans ce chapitre, qui contient nos contributions, nous présentons une technique qui utilise les réseaux de référence pour modéliser et simuler une maison intelligente afin de redonner une vie autonome, dans leur domicile, à des personnes qui devraient normalement être placées dans des établissements spécialisés. Pour cela, nous allons faire appel au paradigme des *nets-within-nets* (présenté dans le chapitre précédent), ou plus précisément, aux réseaux de référence qui sont une implémentation de ce paradigme, augmenté du système expert JESS (Java Expert System Shell).

4.2 Maison intelligente pour les personnes à besoins spécifiques : la spécification

Nous imaginons que cette maison (Figure 4.1) est composée de deux chambres, un hall, d'une salle de réception, d'une cuisine et d'une salle de bain. Etant intelligente, une telle maison doit contenir un large éventail de services qui coopèrent pour simplifier la vie des habitants, de faire des économies d'énergie, et de fournir des solutions de sécurité et de confort. La figure 4.1 montre un exemple d'une telle maison où divers appareils électroménagers sont éparpillés dans les chambres pour constituer un environnement collaboratif fournissant des services intelligents et assurant la sécurité des habitants. Pour cela, plusieurs détecteurs y sont installés : Par exemple, des détecteurs du niveau et du taux de température (Tl) et (Tr). Des détecteurs de fumée (S) pour détecter les incendies, un détecteur de gaz (G) pour détecter le risque de fuite de gaz. Il y a des capteurs d'image (I) et des détecteurs de mouvement (M) dans toutes les chambres afin d'identifier tout visiteur et vérifier s'il est autorisé ou non. Des détecteurs de bris de verre (B) et détecteur de contact (C) sont également installés sur toutes les fenêtres/portes pour vérifier leur innocuité. En outre, la maison intelligente fournit des solutions d'économie d'énergie grâce à l'installation de détecteurs de présence (O) dans toutes les chambres. Le système d'éclairage (L) utilisera les conditions environnementales pour aider à économiser plus d'énergie. Enfin, les chambres devraient être sensibles à la présence humaine. Être sensible implique de connaître les exigences des utilisateurs, apprendre ou connaître leurs préférences, c'est pourquoi nous avons installé dans chaque chambre un agent cognitif stationnaire qui servira les habitants de cette maison, à savoir : la personne âgée, son médecin traitant et un proche (disons son fils).

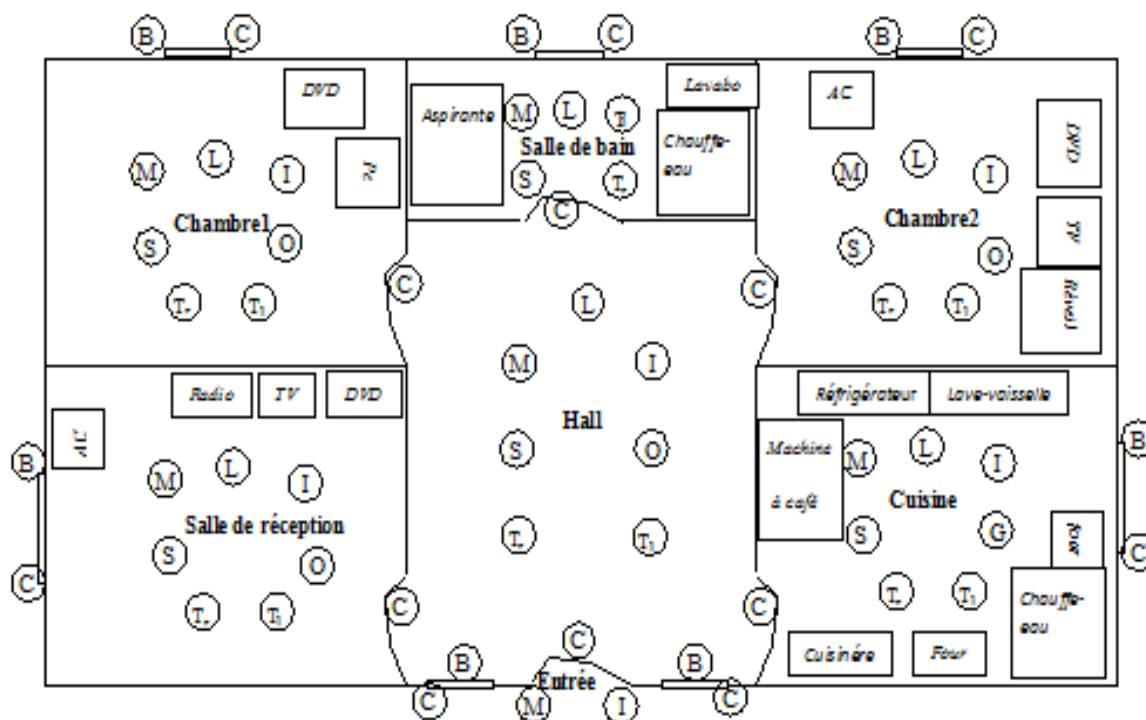


Figure 4.1 : Exemple d'une maison intelligente.

4.3 Modélisation des composants de base de la maison intelligente

Dans cette partie nous allons montrer la manière avec laquelle nous avons modélisé les composants de base de la maison intelligente [83] en se basant sur le paradigme des réseaux dans les réseaux.

4.3.1 Modélisation de l'environnement

Il est souvent indésirable, lors de la modélisation d'un système complexe, de voir toute sa complexité au moment de la modélisation et/ou l'exécution. Par conséquent, la notion de vue est introduite. Plusieurs vues sur un système multi-agent sont possibles. Par exemple le système d'agents, l'ensemble des plateformes, l'agent lui-même ou tout simplement son comportement. L'utilisation des réseaux dans les réseaux comme paradigme de modélisation permet d'exploiter ceci comme suit : l'ensemble du système sera conçu comme un réseau avec les places définissant les chambres : hall, séjour, cuisine, chambre à coucher, chambre 2 et salle de bain. Les transitions modélisent les mouvements possibles entre ces pièces, figure 4.2.

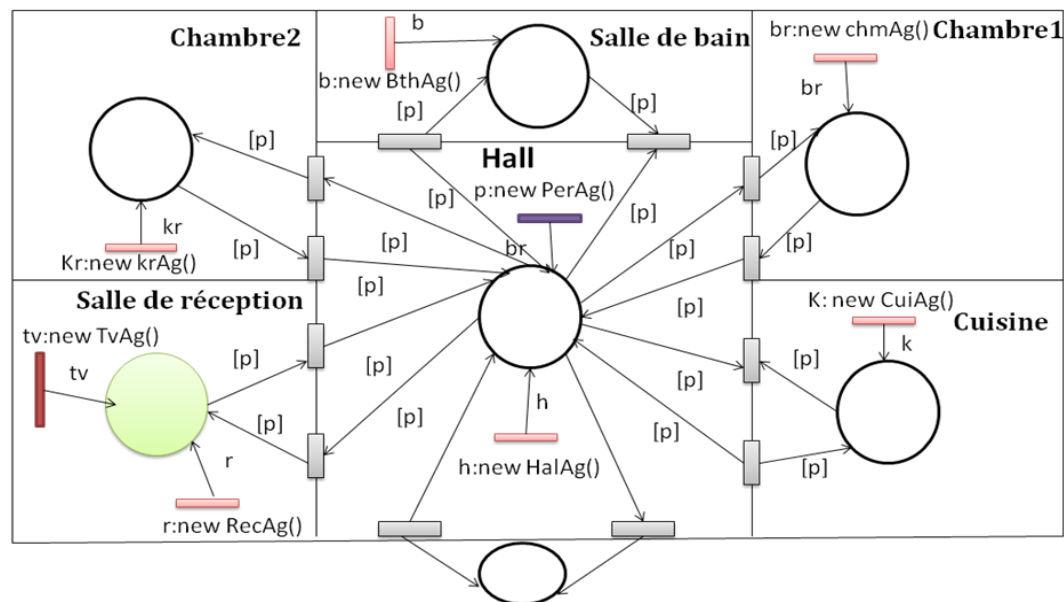


Figure 4.2 : La maison intelligente comme réseau système.

4.3.2 Modélisation des agents

Un agent est une entité qui doit pouvoir recevoir, traiter, et générer des messages. Par rapports aux paradigmes traditionnels, l'agent est doté des concepts nouveaux suivants : l'autonomie, l'intelligence et la mobilité. Dans notre cas, un agent est intelligent puisqu'il a accès à la place « base de connaissance » qui décrit les connaissances de l'agent sur son environnement. Le comportement de l'agent est décrit en termes de protocoles qui se trouvent sous forme de modèles dans la place « protocole ». La sélection des protocoles peut être effectué de façon proactive (autonomie dynamique : la capacité d'engager une action sans appel externe) ou réactive (autonomie déterministe : la capacité de refuser ou de modifier une demande externe). L'arrivée d'un message et le commencement d'une nouvelle conversation sont tous deux influencés par la place « base de connaissance ». Cependant, dans le cas d'une sélection proactive la place base de connaissance est la seule et unique condition de sélection. La sélection et l'activation des protocoles est appelée conversation, car elle implique généralement l'échange de messages.

Tous nos agents partagent la même structure, mais avec des bases de connaissances et des protocoles différents (comportement différents).

4.3.2.1 Cas d'étude 01

Le comportement et les actions du propriétaire de la maison sont étroitement liés à l'heure ou plus précisément à l'heure de la journée dans laquelle il se trouve. Autrement dit, le contexte des agents qui représentent les différents appareils installés à la maison change à chaque intervalle de temps. Intuitivement, ces intervalles de temps sont : réveil, petit déjeuner, sortir de la maison, retour à la maison, déjeuner, dîner et sommeil. De tous les agents dans la maison, nous insisterons en particulier sur les agents installés dans la salle de réception à savoir : la personne âgée, la télévision et l'agent de contexte, figure 4.3. Dans ce qui suit, nous détaillerons ces agents ainsi que les scénarios possibles :

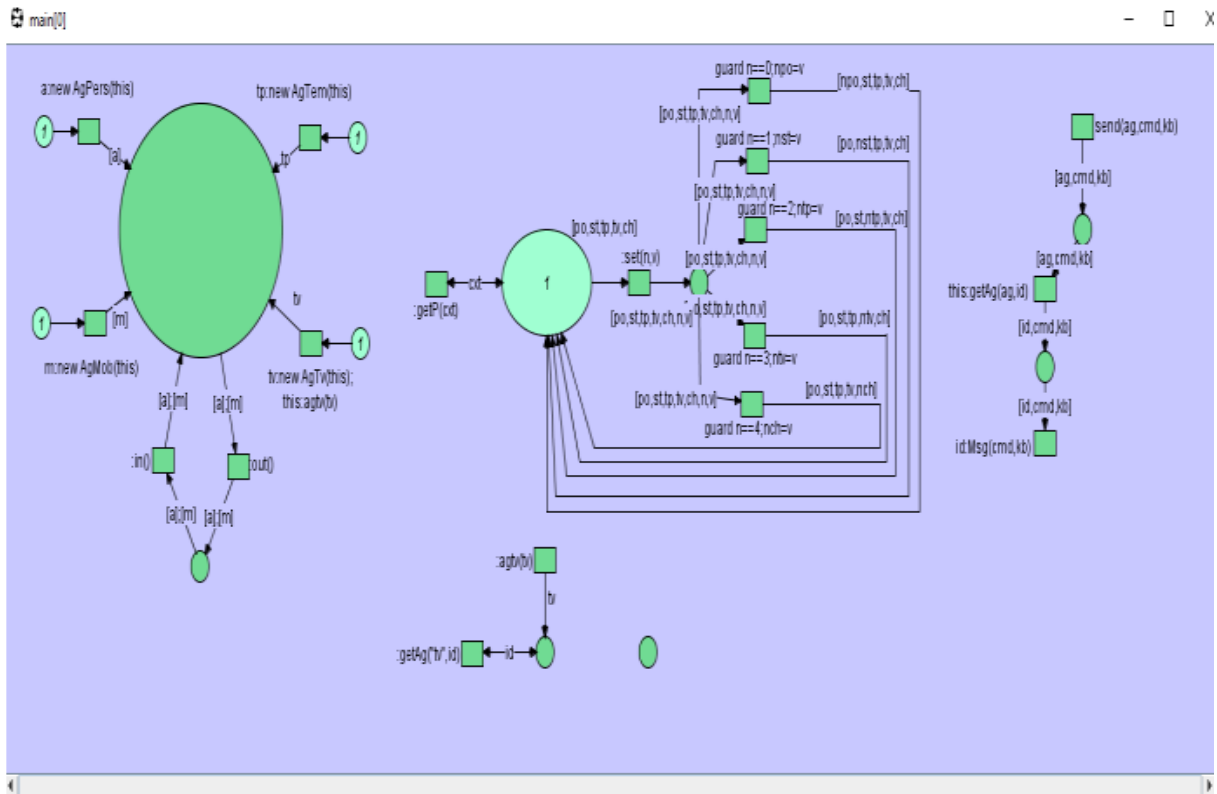


Figure 4.3. : Le réseau principal.

4.3.2.1.a L'agent « contexte » : Cet agent joue un rôle majeur dans le fonctionnement de tous les appareils installés à la maison, il peut être appelé "agent de contexte" car il récupère le contexte de tous les agents installés à la maison et selon les informations récupérées, il envoie un message à tous les appareils pour effectuer les changements nécessaires sur leur comportement. On peut alors dire que l'agent de contexte gère le fonctionnement de tous les appareils. Le réseau principal sur cet agent est sa base de connaissances. De façon répétitive, l'agent contexte déclenche la transition « Get Context » pour obtenir le nouveau contexte, prend les mesures appropriées et envoie des messages aux appareils correspondants. Après cela, la transition « Nouveau contexte » enregistre les renseignements contextuels réels pour les comparer avec ceux qui seront reçus lorsque la transition « Get Context » sera

déclenchée une autre fois. par exemple : si la personne âgée prend son petit déjeuner, alors la télévision va commencer sur une chaîne de sport préférée (l'agent "Contexte" envoie un message à l'agent "TV "pour changer son état en ON et sélectionner la chaîne de sport). La figure suivante montre le résultat de l'exécution de la base de connaissances et d'un protocole de l'agent de contexte (figure 4.4).

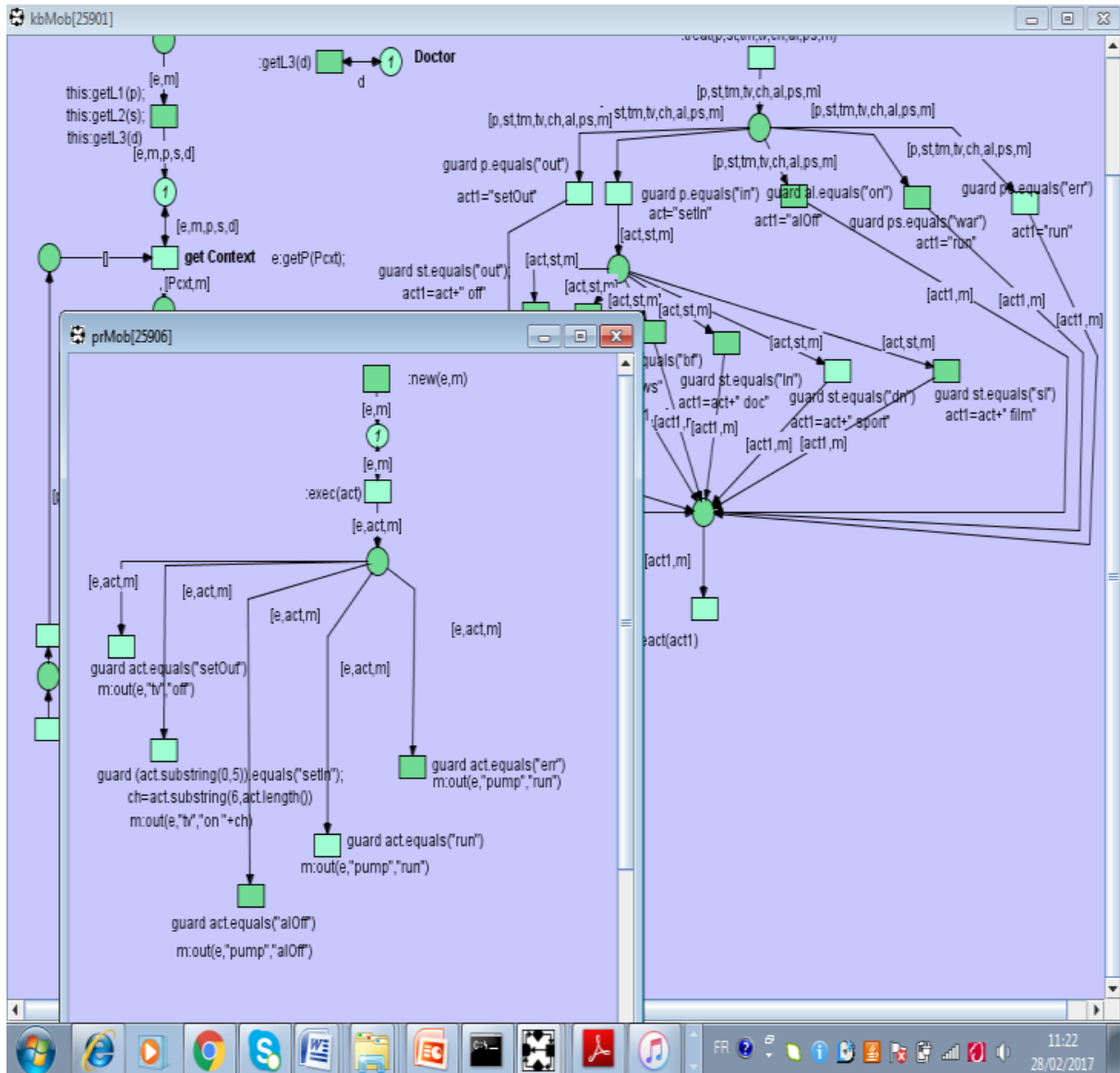


Figure 4.4 : .La base de connaissances et les protocoles de l'agent de contexte.

4.3.2.1.b L'agent « télévision » : Pour la télévision, l'état de la personne joue un rôle clé. Par exemple, si la personne se réveille alors le téléviseur va démarrer sur une chaîne de news préférée (l'agent "contexte" envoie un message à l'agent "TV" pour changer son statut en ON et sélectionner une chaîne de news, et si la personne prend le petit déjeuner alors la télévision va démarrer sur une chaîne de sport préférée (l'agent "contexte" envoie un message à l'agent "TV" pour se positionner sur une chaîne de sport, et si la personne quitte la maison

alors le téléviseur s'éteint (l'agent "contexte" envoie un message à l'agent "TV" pour changer son état en OFF.) Les autres changements qui peuvent se produire sur l'état de la TV ou du canal de télévision sont illustrés dans le tableau suivant (tableau 1) :

Période	Etat de la TV + chaîne
Réveil	On + News channel
petit déjeuner	On + Sport channel
sortir de la maison	Off
retour à la maison	Off
Déjeuner	On + Documentary channel
dîner	On + Movies Channel
sommeil	Off

Tableau 4.1 : Etats possibles de la télévision .

La figure suivante (figure 4.5) montre le résultat de l'exécution des protocoles de l'agent TV :

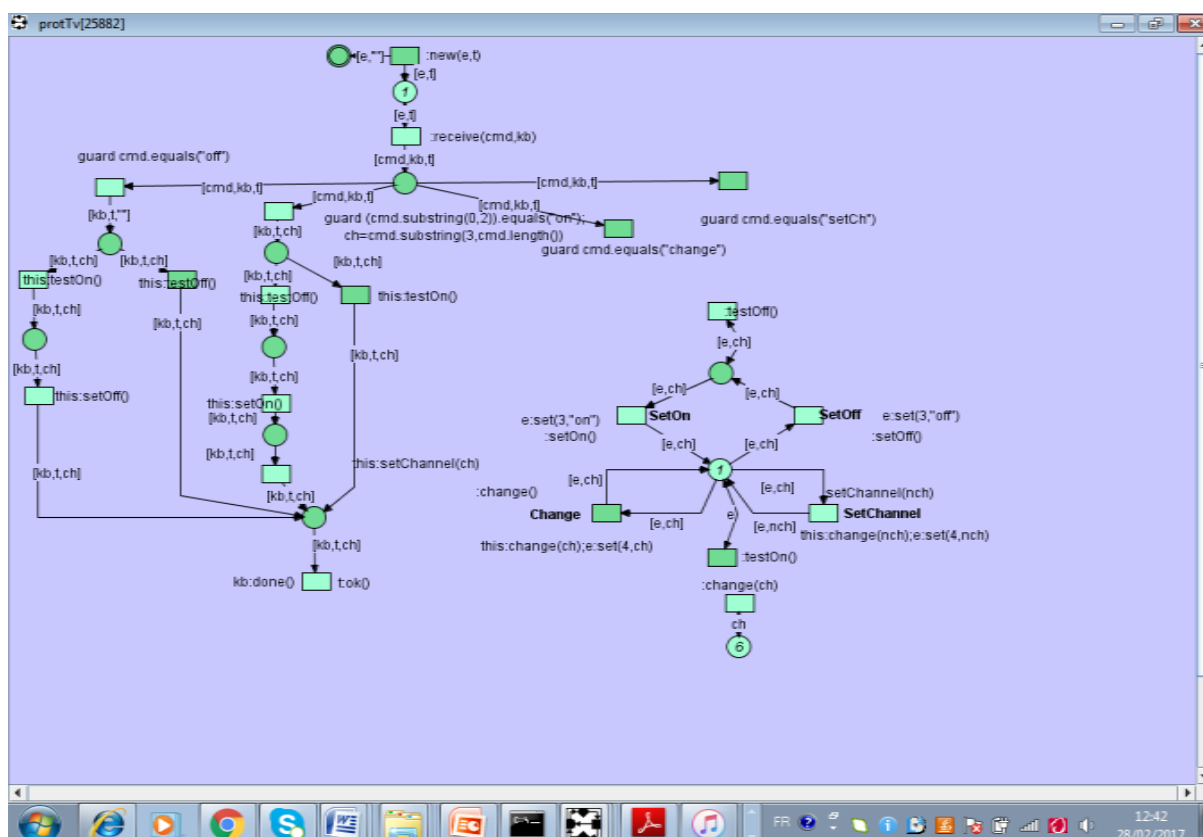


Figure 4.5 : Les protocoles de l'agent TV.

4.3.2.1.c L'agent « personne âgée » : Les périodes de la journée passées par la personne âgée ont été définies dans l'agent de la personne (AgPers) : réveil, petit-déjeuner, sortir de la maison, retour à la maison, déjeuner, dîner et sommeil (Figure 4.6).

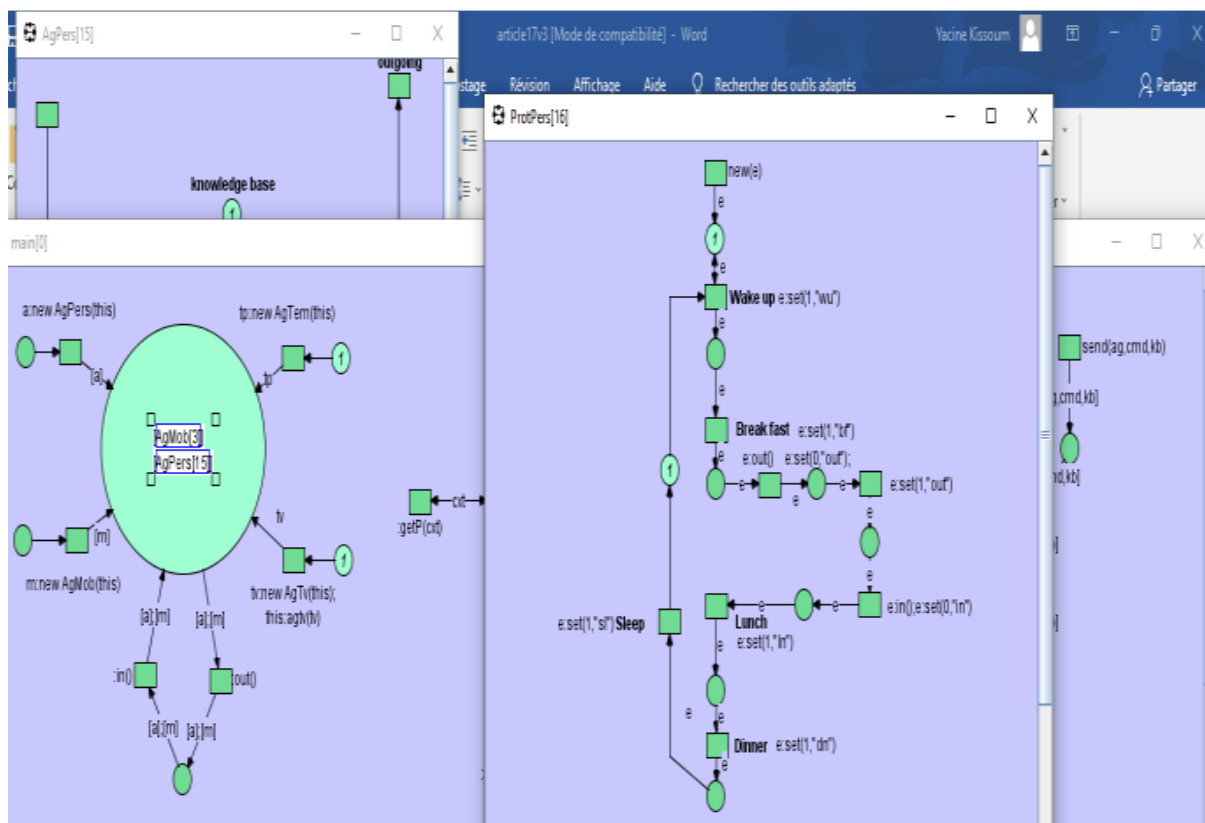


Figure 4.6 : Les protocoles de l'agent personne âgée.

4.3.2.2 Cas d'étude 02

4.3.2.2.a La base de connaissances des agents

Dans les situations simples, la base de connaissances pourrait être représentée par un réseau de Petri. Dans d'autres, plus compliquées, on obtient facilement un réseau de taille importante dont il est difficile de maintenir ou d'apporter la moindre modification. Pour surmonter cette contrainte, nous avons augmenté l'outil de modélisation Renew par une connexion à un système expert, en l'occurrence Java Expert System Shell (JESS).

Un programme écrit en JESS (figure 4.7) consiste en un ensemble de règles, de faits (mémoire de travail) et un moteur d'inférence qui décide quelles règles doivent être exécutées et quand.

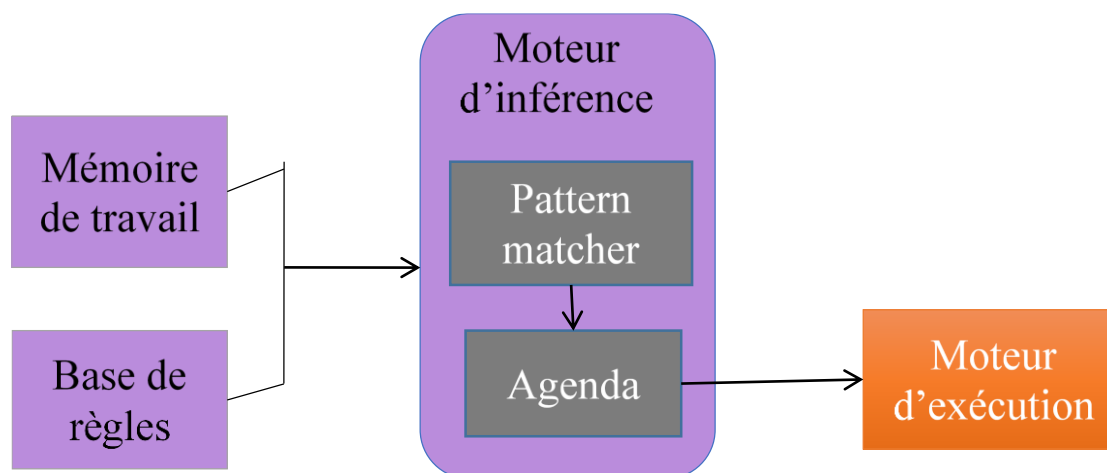


Figure 4.7 : Architecture de JESS.

- **Mémoire de travail** : Ou base de faits, qui contient l'ensemble des connaissances considérées comme vraies. Chaque nouveau fait sera stocké dans cette partie du système expert.
- **Base de règles** : Représente l'ensemble des règles prédéfinies qui seront exécutées par le moteur d'exécution pour inférer de nouveaux faits.
- **Moteur d'inférence** : c'est le module le plus important dans un système expert, il est composé de deux modules :
 - **Pattern matcher** : permet de décider quelle règle exécuter et à quel moment.
 - **Agenda** : Planifie l'ordre dans lequel les règles seront déclenchées.
- **Moteur d'exécution** : Responsable de l'exécution des règles.

Ainsi, nous commençons par définir la mémoire de travail et la base de règles de chaque agent impliqué dans notre maison intelligente. Un intérêt particulier sera accordé à l'agent de la salle de séjour. En effet, un tel agent doit garder un œil attentif sur la personne âgée résidant dans cette maison. Pour cela, il est nécessaire qu'il stocke toutes les informations (appelées aussi faits dans la terminologie JESS) sur cette personne, son environnement ainsi que sur les autres agents. Les faits doivent obéir à un modèle (template). Le modèle a un nom et un ensemble de slots. Les modèles suivants définissent le propriétaire de la maison (la personne âgée) ainsi que les personnes qui lui sont proches :

```
(deftemplate personneAgée (slot prénom) (slot sexe) (slot age) (multislot  
preferTV) (multislot preferMusic) (multislot maladie))
```

```
(deftemplate proche (slot prénom) (slot sexe) (slot age) (slot type) (slot  
telephone))
```

D'autres faits, nécessaires à notre agent de la salle de séjour, sont donnés dans ce qui suit. Ils lui permettent, entres autres, de créer l'ambiance souhaitée par le propriétaire une fois entré dans la salle de séjour.

(deftemplate entrer (slot prenom) (slot pièce) (slot heure))

(deftemplate television (slot etat) (slot chaine))

(deftemplate music (slot etat) (slot genre-music))

(deftemplate rideaux (slot etat))

(deftemplate lumière (slot etat))

Enfin, et étant donné que cette maison est destinée pour une personne âgée, souffrant d'une ou plusieurs maladies (généralement chroniques), l'agent de la salle de séjour doit être en mesure d'analyser les facteurs significatifs en relation avec ces maladies. Pour cela, notre agent utilisera les templates suivants :

(deftemplate pressionArterielle (slot valMax) (slot valMin))

(deftemplate glycémie (slot taux) (slot état))

(deftemplate resultat (slot resultVal))

Assigner des valeurs à l'ensemble des slots de chaque template permet à l'agent stationnaire de la salle de séjour d'identifier la ou les personnes ayant accès à cette maison. Un exemple de personnes autorisées est le suivant :

(assert (personneAgée (prenom omar) (sexe masculin) (age 69) (preferTV Infos Docs Divertissement) (preferMusic Classic Valse) (maladie Diabète Hypertension))

(assert (proche (prenom Ali) (sexe masculin) (age 45) (type Medecin) (telephone 077777777))

(assert (proche (prenom Mohamed) (sexe masculin) (age 45) (type fils) (telephone 06666666))

Maintenant, nous allons définir quelques règles JESS. Chaque règle a deux parties séparées par le symbole "=>". Le côté gauche d'une règle se compose de modèles qui sont utilisés pour faire correspondre les faits dans la mémoire de travail, tandis que la partie droite contient les appels de fonction. Deux exemples de règles sont donnés dans ce qui suit :

Dans la première, aussitôt que le propriétaire entre dans la salle de réception, l'agent va vérifier le temps système (le matin dans notre cas) pour, et de manière concurrente, allumer la télévision et la positionner sur une chaîne de news, ouvrir les stores, et éteindre la lumière. Dans la seconde, notre agent de séjour créera l'ambiance relative à la visualisation d'un film de soirée.

```
1:(defrule regle1
```

```
(entrer {prénom == omar && heure >= 8 && heure <= 10})
```

```
(personneAgée {preferenceTv == info}) =>
```

```
(assert (television (etat allumé) (chaîne Aljazeera)))
```

```
(assert (music (etat éteint)))
```

```
(assert (rideaux (etat ouverts)))
```

```
(assert (lumière (etat éteint)))
```

```
(printout t "heure: entre 8h et 10h | Tv: Aljazeera | music: éteint | rideaux:  
ouverts | lumière: éteint" crlf)
```

```
2:(defrule regle2
```

```
(entrer {prénom == omar && heure > 18 && heure <= 20})
```

```
(personneAgée {preferenceTv == Film
```

```
})=>
```

Chapitre04 : Modélisation des maisons intelligentes pour personnes à besoin spécifiques

(assert (music (etat eteint)))

(assert (rideaux (etat ferme)))

(assert (lumiere (etat allumé)))

(assert (television (etat allumé) (chaine Action)))

(printout t "heure: entre 18h et 20h | Tv: Action | music: éteint | rideaux: fermé
| lumiere: allumé" crlf))

Par ailleurs, les règles suivantes permettent à l'agent de la salle de séjour de statuer sur l'état de santé du propriétaire en fonction de son taux de glycémie.

3 :(defrule regle3

(glycémie {taux < 0.60 }) =>

(assert (resultat (resultVal Hypoglycémie)))

4 :(defrule regle4

(glycémie {taux > 0.70 && taux <1.10 && etat == jeun }) =>

(assert (resultat (resultVal Glycémie normale)))

5 :(defrule regle5

(glycémie {taux > 0.70 && taux <1.10 && etat == repas }) =>

(assert (resultat (resultVal Glycémie normale)))

6 :(defrule regle6

(glycémie {taux > 1.10 })

=>

(assert (resultat (resultVal Hyperglycémie)))

En fonction des faits valides, l'une des règles JESS sera exécutée. L'exécution de la règle permet de sélectionner et déclencher le protocole adéquat de chaque agent.

4.3.2.2.b Les protocoles des agents

Il est inutile d'insister sur le fait que le comportement de n'importe quelle personne chez soi est purement instinctif et fortement dépendant de la tranche horaire dans laquelle il se trouve. Intuitivement, ces tranches de temps sont le matin, midi, l'après-midi et la soirée. Ainsi, différents scénarios rattachés à ces tranches de temps peuvent être imaginés : le « mode réveil », le « mode déjeuner », le « mode sortir », le « mode rentrer », le « mode soirée » et le « mode coucher » sont des exemples de ces scénarios.

A tout instant de la journée, l'agent de séjour tentera de rendre confortable le quotidien du propriétaire de cette maison. Ceci dit, il doit également être en mesure de lui rappeler ses moments de médication, collecter (et par conséquent surveiller) sa tension artérielle et son taux de glycémie à des moments précis et si un problème est détecté, prévenir son médecin traitant et/ou la personne qui lui est proche.

En « mode soirée », par exemple, l'exécution du programme Jess rattaché à la base de connaissance de notre personne âgée (modélisées dans notre cas par un agent mobile), conclura avec un fait valide « voir un film ». Guidé par ce fait, l'agent personne âgée sélectionne puis instancie le protocole : « Entré au salon ». La figure 4.8 montre un tel protocole :

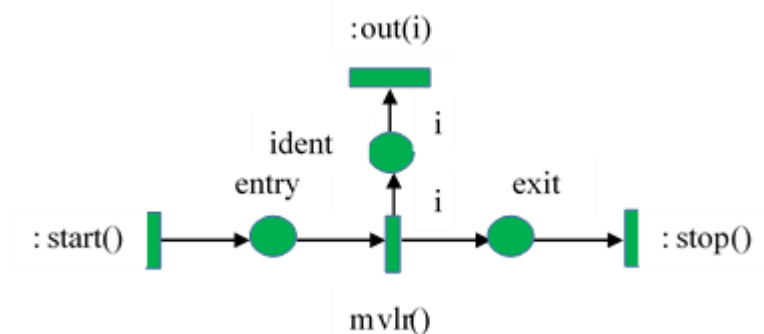


Figure 4.8 : Le protocole entré au salon.

Après l'instanciation de ce protocole, la transition `mvlr()` produit une performative `i` définissant l'identité de la personne accédant au salon qui sera transmise vers l'agent de la salle de séjour via le canal asynchrone `!out(i)`. Ceci fait, le protocole se termine (en exécutant la transition `stop`). La transition `:out(i)` est satisfaite par synchronisation avec la même transition du réseau de l'agent de la salle de séjour. Influencé par l'identité de l'agent entrant au salon, et

les résultats d'une inférence du programme Jess se trouvant dans sa base de connaissances, l'agent de la salle de séjour va instancier le protocole «bienvenu ». Après l'instanciation de ce protocole, et connaissant l'identité et les préférences de la personne entrant dans la salle de séjour, un message de bienvenu est lui est adressé. Par la suite et via la transition *send*, des ordres sont envoyés aux autres agents pour qu'ils créent l'ambiance souhaitée (allumer le téléviseur et fermer les stores).

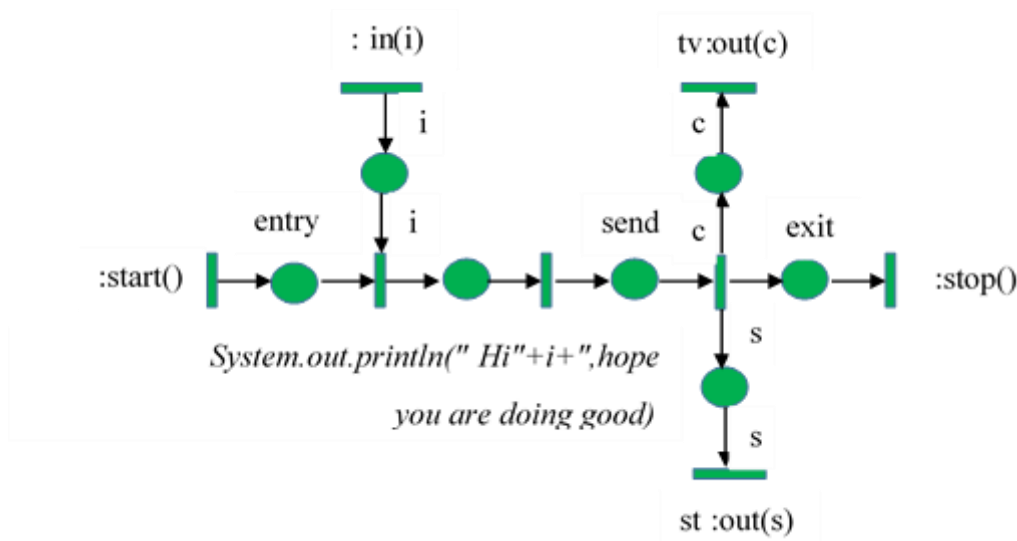


Figure 4.9 : Le protocole bienvenu.

Le dernier protocole concerne la réaction de l'agent de la salle de séjour suite à la réception des valeurs de son taux de glycémie et de sa pression artérielle de la part des agents concernés par ces prélèvements. Notre agent, dans ce cas, comparera ses valeurs à des normes prédéfinies (transition *verify*) et dans le cas défavorable, l'agent exécutera la transition *call doctor*, Figure 4.10

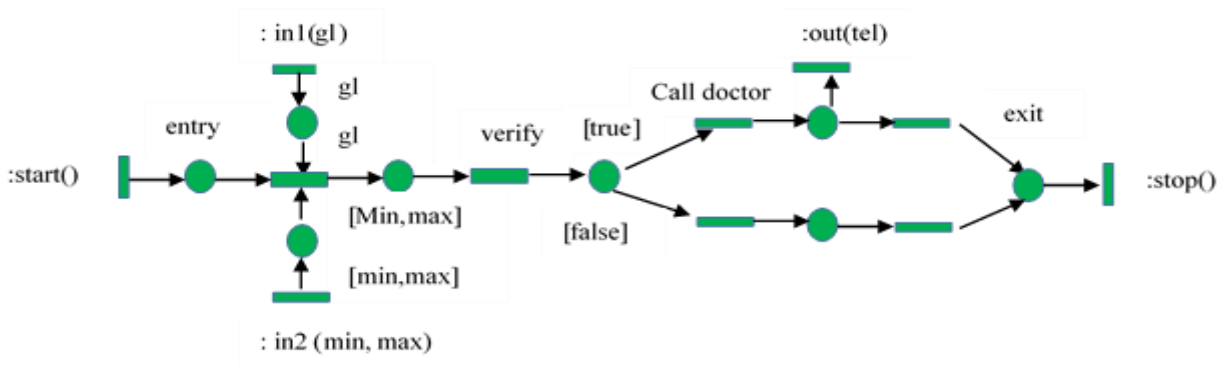


Figure 4.10 Le protocole suivi patient.

Ayant défini les fonctionnalités de notre maison intelligente ainsi que celles des agents concernés, notre système est, à présent, prêt à être exécuter (simuler).

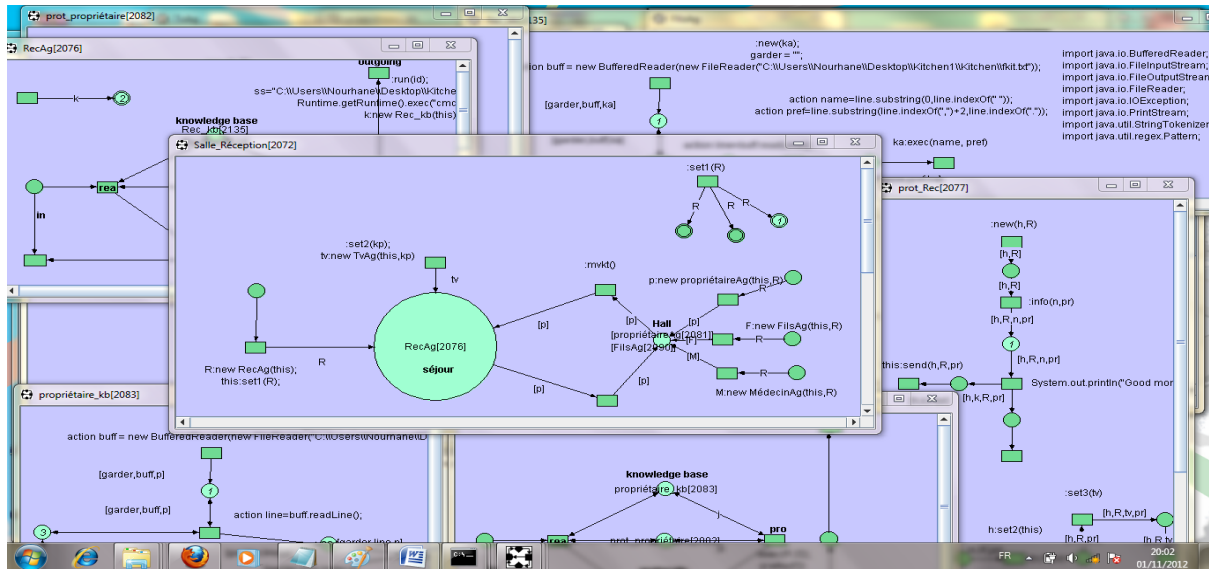


Figure 4.11 : Exemple de Simulation.

La figure 4.11 montre un exemple d'une simulation de notre cas d'étude. Seules les places séjour et hall sont représentées dans cet exemple. Les transitions rattachées à la place séjour sont responsables de la création de l'agent salle de séjour et de l'agent téléviseur. Celles rattachées à la place hall permettent de créer nos agents mobiles : le propriétaire, son fils et son médecin. Les fenêtres en arrière-plan contiennent les réseaux des agents ainsi que leurs protocoles.

4.4 Conclusion

Dans ce chapitre, nous avons présenté deux approches de modélisation d'une maison intelligente destinée aux personnes âgées. Pour cela nous avons fait recours à l'outil Renew qui est un simulateur très puissant et facile à manipuler pour la modélisation des systèmes multi-agent mobiles. Nous avons également utilisé Jess pour représenter les bases de connaissances des différents agents. A notre avis, le paradigme des réseaux dans les réseaux est le plus approprié pour tels système car il est capable d'exprimer les différents types de mobilité des agents et est fondé sur un formalisme dont la sémantique est formelle autorisant, par conséquent, la vérification et l'exécution des systèmes mobiles.

Chapitre04 : Modélisation des maisons intelligentes pour personnes à besoin spécifiques

L'utilisation du paradigme des réseaux dans les réseaux fournit une vue synthétique et permet un contrôle parfait de l'ensemble du système. En d'autres termes, il suffit de cliquer sur un jeton particulier pour que Renew ouvre le réseau schématisant le jeton choisi, permettant ainsi une inspection complète du système pendant la simulation sans aucune interruption. L'avantage est double : (1) en utilisant les réseaux dans les réseaux comme paradigme de modélisation, mène directement à un modèle exécutable du système. Pour obtenir un tel résultat, un processus de modélisation classique nécessitera au moins trois étapes : (a) la modélisation du système, (b) l'implémentation du modèle et (c) l'écriture du programme de visualisation, (2) Visualiser et observer l'exécution du système modélisé n'est autre que la mise en œuvre du système lui-même. Ceci élimine toute possibilité d'introduire des erreurs en passant du modèle à la visualisation des résultats via son implémentation.

Chapitre 05

La pompe à insuline intelligente.

5.1 Introduction

Les parents : nous les avons tous, et pour chacun d'entre nous, il viendra un jour où nous devons veiller à leur bien-être, comme ils ont veillé sur le nôtre. La question se pose alors : comment assurer une bonne qualité de vie à ces proches lorsque leur autonomie s'amenuise ? Il deviendra donc essentiel de continuer à surveiller ces personnes vulnérables dans des environnements sécurisés, en dehors des limites de l'hôpital. Pour cette raison, il serait encore plus urgent de créer des maisons intelligentes pour les personnes âgées ayant perdu leur autonomie et ayant besoin d'aide dans leurs tâches quotidiennes, ainsi que d'une surveillance de la santé.

De plus, la vie autonome, la sécurité et une durée de vie plus longue dans sa propre maison peuvent être améliorées grâce aux technologies de la maison intelligente. La mise en œuvre de telles technologies pourrait améliorer le niveau de vie des personnes tout en réduisant la pression sur les maisons de retraite et autres ressources de soins.

Dans ce chapitre, nous présentons le modèle ACOSO (Agent-based Cooperating Smart Objects), une approche de système multi-agents (MAS) pour l'analyse, la conception et l'implémentation d'une pompe à insuline intelligente pour les personnes âgées souffrant de diabète, fournissant des services de santé, de confort et de sécurité tout en surveillant les activités quotidiennes de l'utilisateur.

5.2 La pompe à insuline:

Un des problèmes de santé majeurs pour les personnes âgées atteintes de diabète survient lorsque le corps ne produit pas assez d'insuline. Cela entraîne souvent des complications telles que l'amputation, une insuffisance rénale, la cécité et/ou des problèmes cardiaques. Selon des études récentes, les risques associés au diabète peuvent être réduits lorsque les niveaux de glucose dans le sang sont régulièrement surveillés et gérés. Il est donc essentiel de surveiller de façon constante et efficace les taux de sucre dans le sang des patients diabétiques.

Les améliorations apportées à cette discipline ont permis d'améliorer les soins et la qualité de vie des personnes atteintes de diabète. Le risque d'hypoglycémie [81], l'effet secondaire le plus craint de la thérapie à l'insuline chez les patients, les soignants et les

médecins, rend difficile l'atteinte d'un contrôle glycémique serré, même avec une insulinothérapie intensive et des régimes d'insuline contemporains.

En associant la surveillance sous-cutanée continue du glucose à l'administration d'insuline par voie sous-cutanée au moyen d'une pompe à insuline [56], on peut créer des systèmes en boucle fermée qui délivrent de l'insuline en continu en fonction des niveaux de glucose [5]. Les prototypes actuels utilisent la méthode sous-cutanée pour mesurer les niveaux de glucose interstitiel et administrer l'insuline. La pompe à insuline est un dispositif thérapeutique qui infuse en continu une quantité prédéfinie d'insuline à action rapide à un débit constant par l'intermédiaire d'un cathéter [15].

Plus précisément, la pompe à insuline est un petit appareil ressemblant à un téléphone mobile (5 x 8 x 1,9 cm et pesant moins de 100 g en moyenne) qui fonctionne sur batterie. Il est constitué de plusieurs composants, comme le montre la figure 5.1:



Figure 5.1: Composant de la pompe d'insuline.

- Une boîte contenant un réservoir rempli d'insuline ultra-rapide;
- Des composants électroniques permettant un contrôle précis de l'insuline fournie par la pompe;
- Un tube mince (allant de 60 cm à 1,1 m, selon le modèle) relié au réservoir par un connecteur et se terminant par une canule flexible qui pénètre légèrement dans la peau.

•L'aiguille ou la canule est maintenue en place avec un adhésif résistant à l'eau. La canule et le tube forment ensemble le cathéter. La pompe à insuline est indiquée pour certains patients diabétiques qui répondent à des critères spécifiques :

- Recevoir au moins 3, 4 ou même 5 injections d'insuline par jour et effectuer au moins 3 auto mesures de glycémie par jour;
- Avoir une hémoglobine glyquée supérieure à 7 %, ou avoir 2 épisodes hypoglycémiques graves par an et/ou au moins 4 épisodes hypoglycémiques "modérés" par semaine, ou avoir un taux de glycémie instable après 6 mois de traitement;
- Avoir des besoins en insuline flexible en raison de facteurs physiologiques, du mode de vie ou de conditions telles que les infections chroniques, les neuropathies douloureuses ou l'intolérance à l'injection.

5.3 Spécification, analyse, conception et implémentation du système

5.3.1 Spécification du système

Une pompe à insuline est un système conçu pour administrer des doses régulières d'insuline aux patients diabétiques. Le système comprend plusieurs composants, comme indiqué dans le schéma de principe de la figure 5.2. Les petites cases marquées du « s » représentent des capteurs. Les spécifications de chaque composant sont énumérées ci-dessous :

- **Needle assembly:** Connecté à la pompe, l'assemblage de l'aiguille est responsable de l'administration d'insuline dans le corps du patient.
- **Sensor :** Le capteur mesure la glycémie du patient.
- **Pump :** La pompe achemine l'insuline du réservoir vers le jeu d'aiguilles.
- **Controller :** Le contrôleur est équipé d'un interrupteur à trois positions (arrêt/auto/manuel) et d'un bouton permettant de régler le nombre d'unités d'insuline à administrer (1 unité par pression). Le contrôleur est responsable de la gestion de l'ensemble du système.
- **Alarm :** L'alarme est déclenchée lorsqu'il y a un problème avec le système.
- **Displays :** Le système comporte trois affichages, représentés par les variables display1, display2 et clock. L'afficheur1 affiche les messages du système, l'afficheur2 la dernière dose d'insuline administrée et l'horloge indique l'heure actuelle.

•**Clock** : L'horloge fournit au contrôleur l'heure actuelle. Pour des raisons de sécurité, les utilisateurs ne peuvent pas modifier l'horloge.

•**Transceiver** : L'émetteur-récepteur permet de transmettre et de recevoir des informations et des commandes à l'infirmière ou au médecin, et d'en recevoir de l'autre côté.

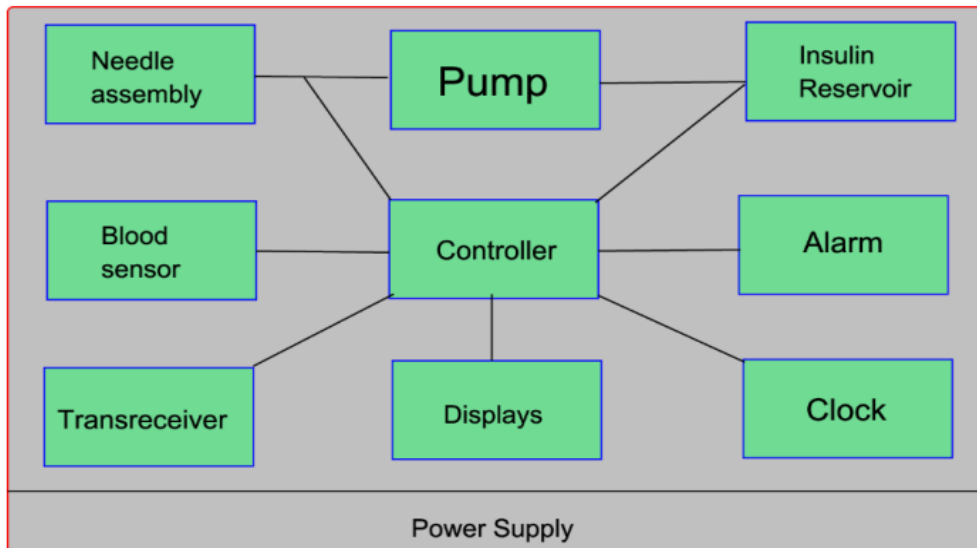


Figure 5.2 .Architecture matérielle de la pompe à insuline intelligente.

5.3.2 Analyse du système

La pompe à insuline intelligente est un objet intelligent (SO) physiquement basé sur le logiciel de commande de la pompe. Il fait partie d'une maison intelligente conçue pour une personne âgée [82]. Grâce à cette pompe, le patient peut gérer son état en utilisant le service SmartHealth [6] qui lui permet de mesurer sa glycémie et d'administrer de l'insuline si nécessaire. Il peut également transmettre les mesures quotidiennes à un médecin, qui pourra surveiller les données et intervenir en cas de problème grave.

Le service SmartHealth (figure 5.3) indique que la pompe démarre après avoir été mise en marche par l'opération StartUp, qu'elle fonctionne automatiquement par l'opération automatique et qu'elle cesse de fonctionner en raison d'une erreur ou parce que l'utilisateur a sélectionné le mode d'opération manuel. Dans ce mode, un bouton est nécessaire pour spécifier la quantité d'insuline à injecter.

Le comportement du système est illustré par l'opération StartUp. Il est important de noter que la dose cumulative n'est pas réglée au démarrage, mais qu'elle est réinitialisée à

zéro à minuit. Cela garantit que le système garde toujours une trace de la dose cumulative totale délivrée, indépendamment des actions de l'utilisateur lors de la mise en marche et de la mise hors tension de la machine.

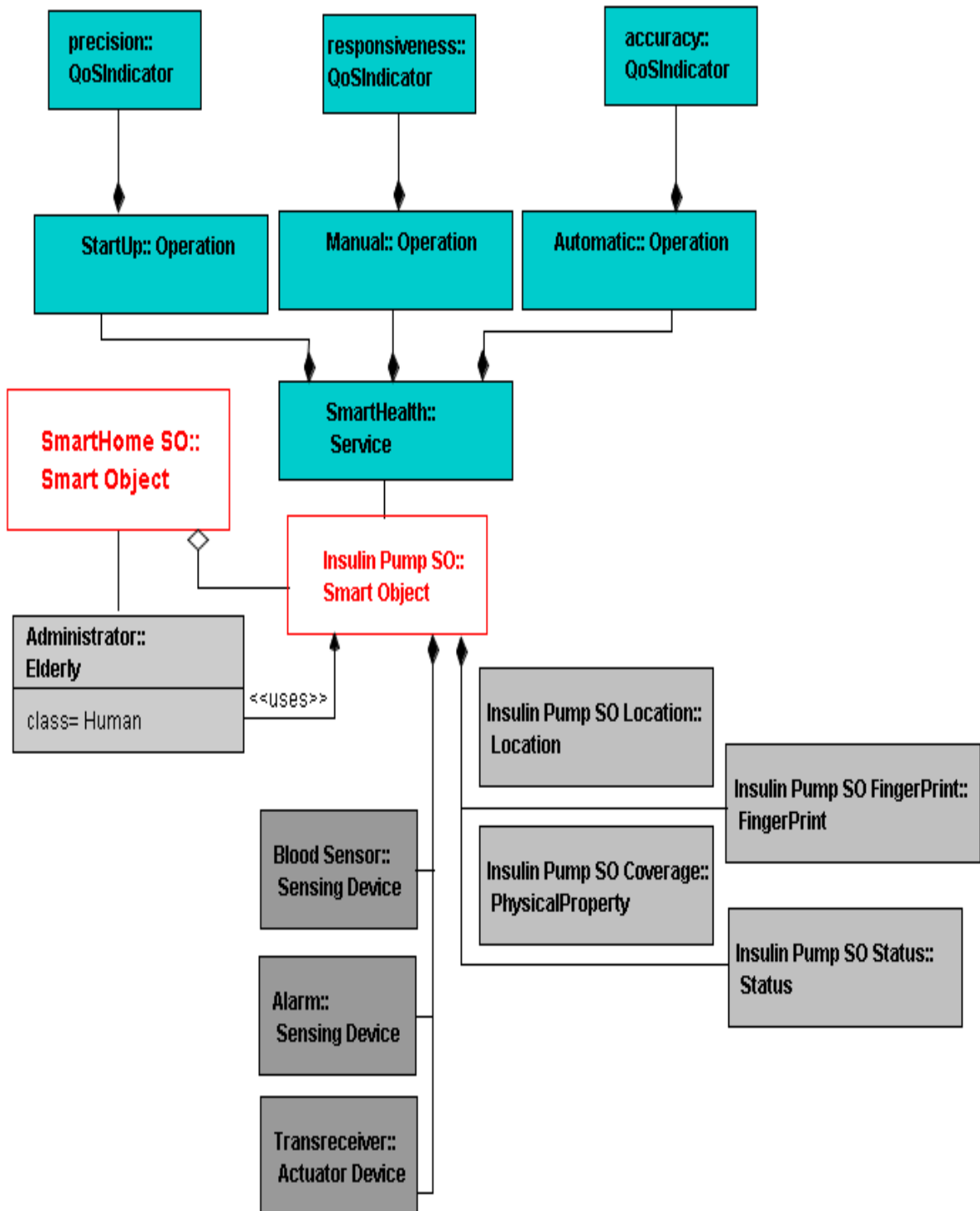


Figure 5.3 : Le modèle de SO de haut niveau à la phase d'analyse.

En mode automatique, le système de pompe à insuline fonctionne en mesurant le taux de glucose sanguin de l'utilisateur toutes les 10 minutes et en administrant de l'insuline si nécessaire. La dose d'insuline requise est déterminée par le taux de glucose actuel, tel que détecté par le capteur. Si le taux de glucose tombe en dessous du minimum sécuritaire, on n'injecte pas d'insuline. L'insuline n'est administrée que lorsque le taux de sucre dans le sang est élevé ou augmente à un rythme accéléré. Inversement, si le taux de glucose dépasse la fourchette recommandée, l'insuline est administrée à moins que le taux de glucose ne soit déjà en baisse et que le taux de décroissance soit stable ou en diminution.

Il est à noter que la quantité réelle d'insuline administrée peut différer de la dose calculée en raison de diverses contraintes. Le système impose une limite à la quantité maximale d'insuline qui peut être administrée en une seule injection, ainsi qu'un plafond sur la dose cumulative totale dans une période de 24 heures. La quantité cumulée est remise à zéro à minuit, ce qui permet au système de suivre avec précision l'insuline totale administrée.

Le méta-modèle SO de haut niveau ACOSO-Meth (Figure 3) introduit des fonctionnalités supplémentaires pour soutenir pleinement la phase d'analyse de l'SO. Ces fonctionnalités décrivent à la fois les aspects statiques et dynamiques. Plus précisément, le système de pompe à insuline est modélisé en utilisant divers concepts, tels que les objets de service (SO), l'information sur les services de base, le service de SO, l'utilisateur de SO, l'augmentation et l'agrégation de SO.

•**SO BasicInfo** fournit des informations essentielles sur le système de pompe à insuline, y compris l'état actuel du système, la position géophysique, les propriétés physiques, l'identificateur, le type et les paramètres de qualité de service (QSo). La pompe à insuline SO a aussi un créateur, et ses utilisateurs peuvent être des humains, des objets intelligents ou des systèmes numériques.

•**Le service SO** est un service en ligne offert par la pompe à insuline SO. Chaque service a un type, un nom, une description et divers types de paramètres d'entrée et de retour, et il est mis en œuvre par l'intermédiaire d'une ou plusieurs opérations. De plus, des indicateurs QSo peuvent être associés à chaque service.

• **L'utilisateur SO** identifie les entités qui interagissent avec les services SO de la pompe à insuline. Ces entités peuvent être des humains, des objets intelligents ou des systèmes numériques.

• **La fonction d'amélioration** désigne les fonctionnalités matérielles et logicielles qui permettent de développer et de rendre intelligent le système de pompe à insuline. L'appareil peut être spécialisé comme ordinateur, capteur ou actionneur.

• **SO Aggregation** facilite l'agrégation de plusieurs SO, permettant une intégration physique et logique. Cela permet de fournir des services plus avancés et cohérents dans les systèmes complexes, comme les bâtiments intelligents.

5.3.3 Conception du système

Le méta-modèle de SmartInsulinPump de haut niveau est perfectionné au cours de la phase de conception, ce qui donne un méta-modèle SmartInsulinPump basé sur ACOSO, comme illustré à la figure 5.4. Plus précisément, le SmartInsulinPump est conçu comme un agent utilisant ACOSO, les utilisateurs de SO agissant en tant qu'agents génériques. Le service SmartHealth et les opérations liées à StartUp, Automatic, et le mode manuel sont modélisés comme UserDefinedTasks (e.g., SmartHealthTask, StartUpTask, AutomaticTask, ManualTask, SugarLowTask, SugarHighTask, and SugarOkTask), déclenchées par les ServiceEvents correspondants (SmartHealthEvent, StartUpEvent, AutomaticEvent, ManualEvent, SugarLowEvent, SugarHighEvent et SugarOkEvent).

En outre, la fonction BloodSensingEvent permet à SmartInsulinPump de s'interfacer avec un capteur de glycémie qui mesure le taux de glycémie actuel. La fonction AlarmEvent signale à l'utilisateur un problème sonore. Il alerte également le médecin par l'intermédiaire d'un dispositif émetteur-récepteur, ce qui lui permet de prendre les mesures appropriées.

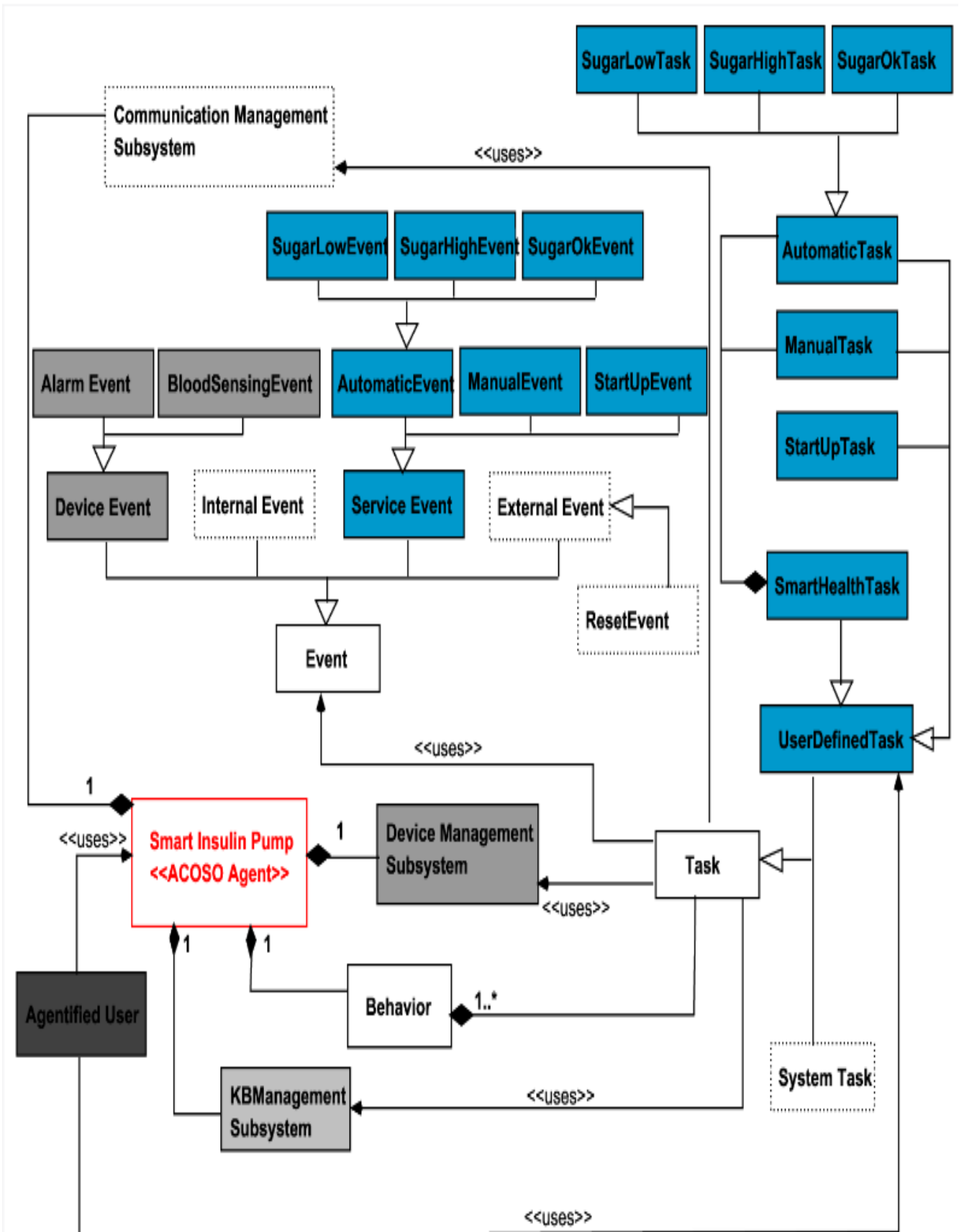


Figure 5.4 : Le méta-modèle de la pompe à insuline intelligente de haut niveau pendant la phase de conception.

Les entités de méta modélisation SO basées sur ACOSO sont regroupées en quatre catégories. La première catégorie est SO Basic Info, qui comprend des informations sur le SO et est divisée entre les entités de méta-modélisation du SO et le sous-système KBManagementSubsystem. Ce dernier gère les données sur le statut global actuel de SO, les déductions et autres informations qui peuvent être envoyées entre les missions. Le deuxième groupe est SO Service, qui se compose des services fournis par le SO. Ces services sont inclus dans des tâches particulières au niveau de l'application, sont personnalisables, faciles à programmer et communiquent avec d'autres composants SO en utilisant ServiceEvents. Le troisième groupe est l'augmentation, qui comprend la commande des dispositifs, des actionneurs et des capteurs inclus dans le SO. Les DeviceEvents sont utilisés pour toutes interactions avec ces dispositifs, indépendamment de leur protocole ou spécificité. Enfin, la quatrième catégorie est SO Communication, qui offre une interface universelle pour la communication avec le SO, à la fois en interne (en utilisant InternalEvents) et en externe (en utilisant ExternalEvents).

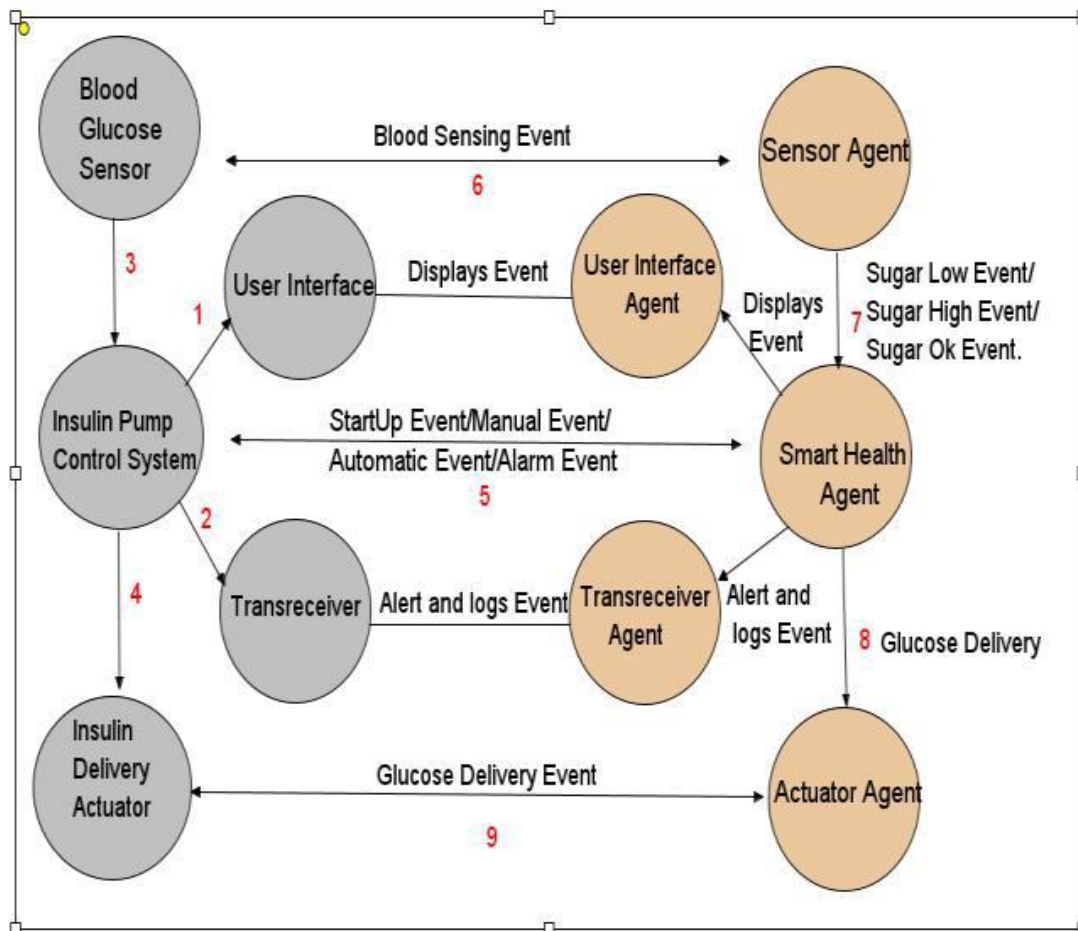


Figure 5.5 : Organigramme des interactions entre les systèmes de Contrôle.

Plus précisément, le système conçu assure une surveillance continue du glucose grâce à l'utilisation d'un capteur de glycémie. Ce capteur fonctionne en temps réel et est configuré pour mesurer les niveaux de glycémie toutes les 10 minutes à l'aide de l'agent de détection, qui s'interface directement avec le matériel du capteur. L'agent capte le niveau de glucose et le communique au système de contrôle de la pompe à insuline par des protocoles de passage de messages asynchrones.

D'autre part, le mécanisme de distribution de l'insuline est réagi par l'agent actionneur qui interagit avec l'actionneur de la pompe à insuline. Le système de contrôle de la pompe à insuline utilise un algorithme de prise de décision qui analyse le niveau de glucose entrant. Selon les seuils préconfigurés et les niveaux de glucose mesurés récemment, le système de contrôle ne fournit pas d'insuline lorsque le niveau de glucose est dans la plage de sécurité ou en dessous. Inversement, il administre des dosages précis d'insuline lorsque le taux de glucose dépasse le seuil de sécurité ou augmente rapidement.

Enfin, le système utilise un suivi cumulatif de la posologie qui se réinitialise à minuit pour assurer la conformité aux limites de sécurité sur une période de 24 heures.

L'architecture du système est construite sur un cadre basé sur les agents, où chaque agent remplit un rôle spécifique au sein de l'écosystème (figure 5.5).

- **Sensor Agent** : Lit les données de glucose du capteur et communique les valeurs au système de contrôle.

- **Control System Agent** (SmartHealth Agent) : agit comme décideur central. Il traite les données des capteurs, applique des règles d'inférence et décide de l'administration de l'insuline. Cet agent utilise des règles prédéfinies stockées dans la tâche de règle d'inférence, qui combinent les lectures de glucose en temps réel avec des données historiques pour prendre des décisions autonomes. Par exemple, si le taux de glucose chute rapidement, l'administration d'insuline est interrompue, même si la valeur du glucose est initialement élevée. Il est important de noter que le système passe en mode manuel en cas d'erreurs ou de dysfonctionnements, ce qui permet aux utilisateurs de remplacer et de spécifier manuellement les dosages d'insuline.

- **Actuator Agent** : Reçoit les instructions de dosage d'insuline de l'agent du système de contrôle et les exécute via le mécanisme de distribution d'insuline.

•**Transceiver Agent** : Assure la communication avec les entités externes (p. ex., le système du médecin) en transmettant des journaux quotidiens et des alertes lorsque les lectures de glucose sont hors de portée.

•**User Interface Agent** : Agit comme intermédiaire entre l'utilisateur et le système, permettant à l'utilisateur d'interagir avec le système, de surveiller son état et d'effectuer des ajustements si nécessaire.

Les agents communiquent au moyen d'un système de messagerie ACL, supporté par la plateforme JADE. Cette configuration assure l'interopérabilité et prend en charge la livraison de messages fiable et asynchrone.

5.3.4 Implémentation du système

Le méta-modèle SmartInsulinPump basé sur ACOSO est affiné au cours de la phase d'implémentation, ce qui donne lieu au méta-modèle SmartInsulinPump basé sur JACOSO (figure 5.6). Plus précisément, le SmartInsulinPump générique agencé agit en tant qu'expert, travaillant avec les utilisateurs agencés de SO et l'agent basé sur JADE. L'adaptateur de communication ACLCommunication permet à la pompe SmartInsulinPump de faciliter un système d'échange de messages direct basé sur ACL. La SmartInsulinPumpInferenceRuleTask comprend à la fois les règles d'inférence requises pour le processus de prise de décision de la SmartInsulinPump et les valeurs actuelles des variables qui composent ces règles. UserDefinedTasks (SmartHealthTask, StartUpTask, AutomaticTask, ManualTask, SugarLowTask, SugarHighTask, and SugarOkTask), qui mettent en œuvre SmartHealth et les événements connexes (StartUpEvent, AutomaticEvent et ManualEvent), sont conçus comme des comportements JADE. En attendant, le SpineAdapter fournit l'interface entre la SmartInsulinPump et ses appareils.

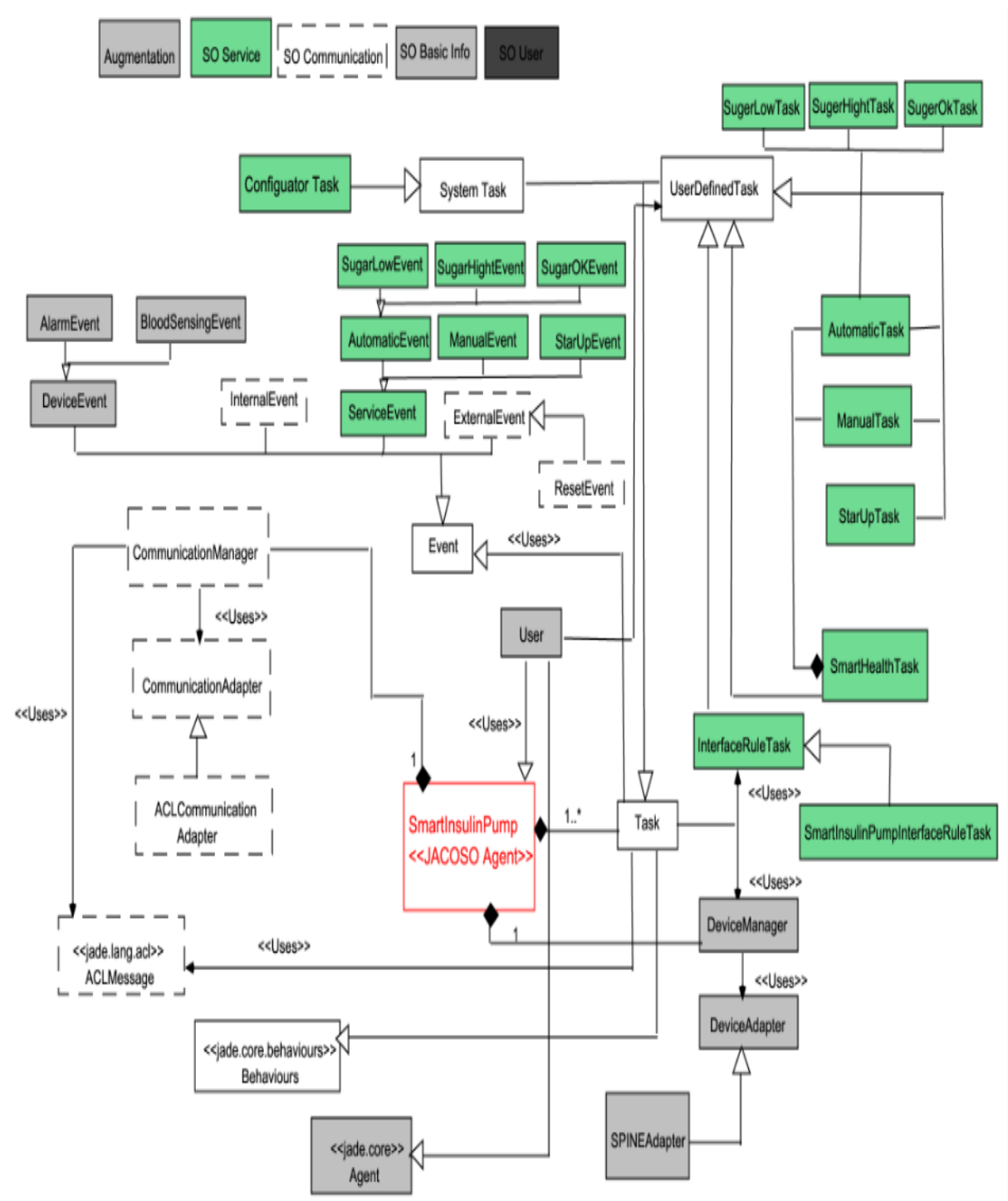


Figure 5.6. Le méta-modèle de la pompe à insuline intelligente à la phase d'implémentation.

5.4 Mécanismes de sécurité et protocoles de secours

Pour assurer la sécurité, la fiabilité et la résilience du système de pompe à insuline intelligente, un ensemble de mécanismes de sécurité et de protocoles de secours doit être pris en compte. Ces mécanismes permettent de remédier aux pannes potentielles des

capteurs, des actionneurs et des systèmes de communication, en veillant à ce que l'appareil continue de fonctionner efficacement même dans des conditions défavorables. Les mécanismes de sécurité comprennent la surveillance continue de l'état des capteurs pour détecter les erreurs ou les incohérences dans les lectures de glucose. Dans les cas où des anomalies sont détectées, le système passe automatiquement en mode manuel et émet des alertes à l'utilisateur et au personnel médical. De plus, pour prévenir les surdoses d'insuline, le système applique des limites supérieures aux doses quotidiennes cumulées et aux doses uniques. Cette fonction sert de protection contre les erreurs de prise de décision ou les dysfonctionnements matériels. Enfin, une fonction de surveillance en temps réel des batteries permet d'alerter rapidement les utilisateurs lorsque la puissance est faible. Avant qu'un arrêt complet ne se produise, la pompe passe à un état de sécurité, ce qui réduit le risque de pannes critiques.

Les protocoles de secours complètent ces mécanismes de sécurité en assurant le fonctionnement continu pendant des problèmes imprévus. Si des défaillances ou des anomalies sont détectées, le système passe en mode manuel, permettant aux utilisateurs d'entrer les doses d'insuline manuellement via l'interface physique. Les alertes d'erreur en temps réel sont transmises au personnel médical par l'intermédiaire de l'agent du transmetteur, ce qui permet une intervention rapide lorsque nécessaire. De plus, le système stocke les données historiques sur le glucose pour générer un calendrier provisoire d'administration de l'insuline pendant les temps d'arrêt des capteurs, ce qui permet de maintenir un certain degré de fonctionnalité et de réduire les risques jusqu'à ce que la surveillance en temps réel puisse reprendre. Les mécanismes de sécurité et les protocoles de secours permettent de relever des défis prévus et imprévus. Ces mesures non seulement améliorent la robustesse du système, mais garantissent également que la sécurité de l'utilisateur ne soit pas compromise dans diverses conditions d'exploitation. En intégrant ces caractéristiques, le système démontre son engagement envers une technologie de soins de santé fiable et centrée sur l'utilisateur.

5.5 Le pseudo-code de la smart InsulinPump pour le processus décisionnel

Le pseudo-code suivant, figure 7, fournit une représentation structurée du processus de prise de décision dans le système de pompe à insuline intelligente. Il montre la dépendance du système à une base de connaissances pour analyser les lectures des capteurs

de glucose, appliquer des règles d'inférence prédéfinies et déterminer le dosage approprié d'insuline.

```
// Initialization

Initialize glucose_threshold_low = 70
Initialize glucose_threshold_high = 180
Initialize max_single_dose = 10 // Maximum insulin per injection
Initialize max_daily_dose = 50 // Maximum total insulin in a 24-hour period
Initialize cumulative_dose = 0
Load knowledge_base = {rules, conditions, actions}
While system_status == "Active":
  Read glucose_level from Sensor Agent
  // Fail-safe: Check Sensor Health
  If Sensor Agent reports error:
    Transition to Manual Mode
    Alert user and doctor
    Continue
  // Retrieve Decision Rules from Knowledge Base
  decision_rules = knowledge_base["rules"]
  // Infer Action Based on Glucose Level
  If glucose_level < glucose_threshold_low:
    Action = "No insulin"
    Log "Glucose level below threshold, no action required"
  Else If glucose_level >= glucose_threshold_low AND glucose_level <= glucose_threshold_high:
    Action = "Maintain current state"
    Log "Glucose level within safe range, no insulin needed"
  Else:
    // High glucose level detected
    dosage = knowledge_base["actions"].calculate_dosage(glucose_level, rate_of_change)
    // Apply Constraints
    If dosage > max_single_dose:
      dosage = max_single_dose
```

```

If (cumulative_dose + dosage) > max_daily_dose:
    dosage = max_daily_dose - cumulative_dose

If dosage > 0:
    Action = "Inject insulin"
    cumulative_dose += dosage
Else:
    Action = "No insulin"
    Log "Daily dose limit reached, no insulin administered"

// Execute Action
If Action == "Inject insulin":
    Actuator Agent administer(dosage)
    Log "Injected insulin dosage: ", dosage
    
```

Figure 5.7 : Le pseudo-code de la smart InsulinPump pour le processus décisionnel.

5.6 Diagramme de séquence

Le diagramme de séquence (figure 5.8) illustre le service SmartHealth du SmartInsulinPump :

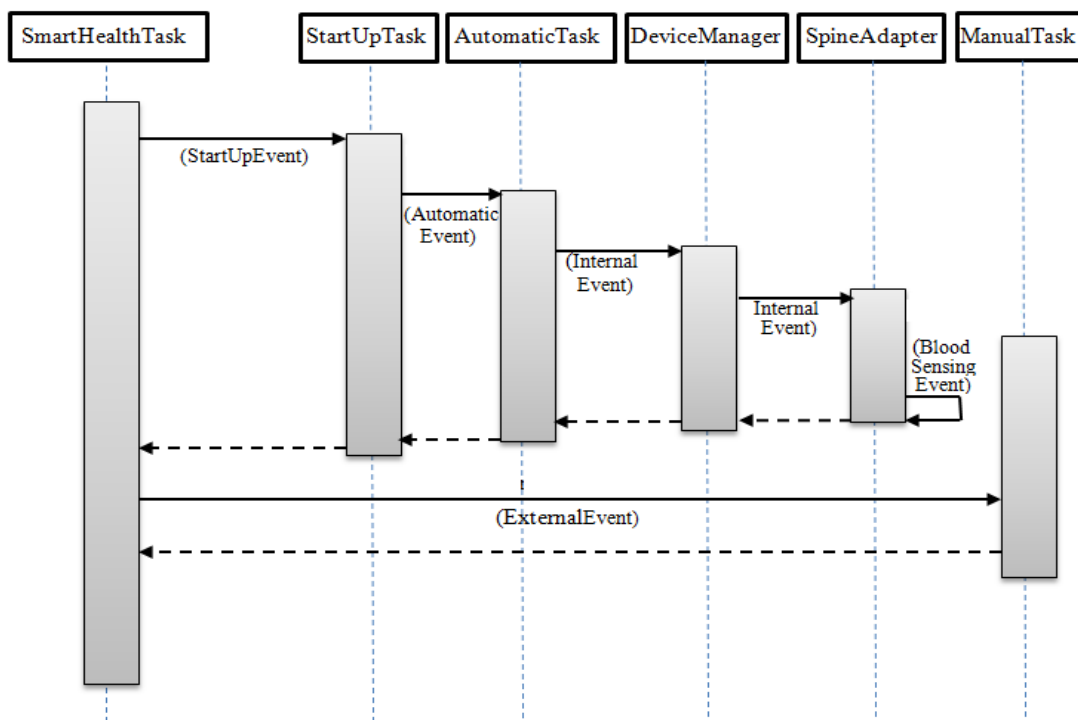


Figure 5.8 : Diagramme de séquence illustrant le fonctionnement de la pompe à insuline intelligente.

5.7 La simulation de la pompe d'insuline

Un prototype pour la simulation de la pompe à insuline et d'autres services dans la maison intelligente pour les personnes âgées a également été développé en utilisant la plateforme JADE (figure 5.9). Un environnement de simulation a été créé pour évaluer la fonctionnalité du système dans diverses conditions, comme les fluctuations des taux de glycémie, les interactions avec l'utilisateur et les anomalies potentielles. Bien que les résultats de validation spécifiques ne soient pas encore disponibles, les travaux futurs seront axés sur l'évaluation des mesures clés du rendement, y compris la précision, le temps de réponse et le traitement des erreurs, par des tests détaillés dans des scénarios réels.

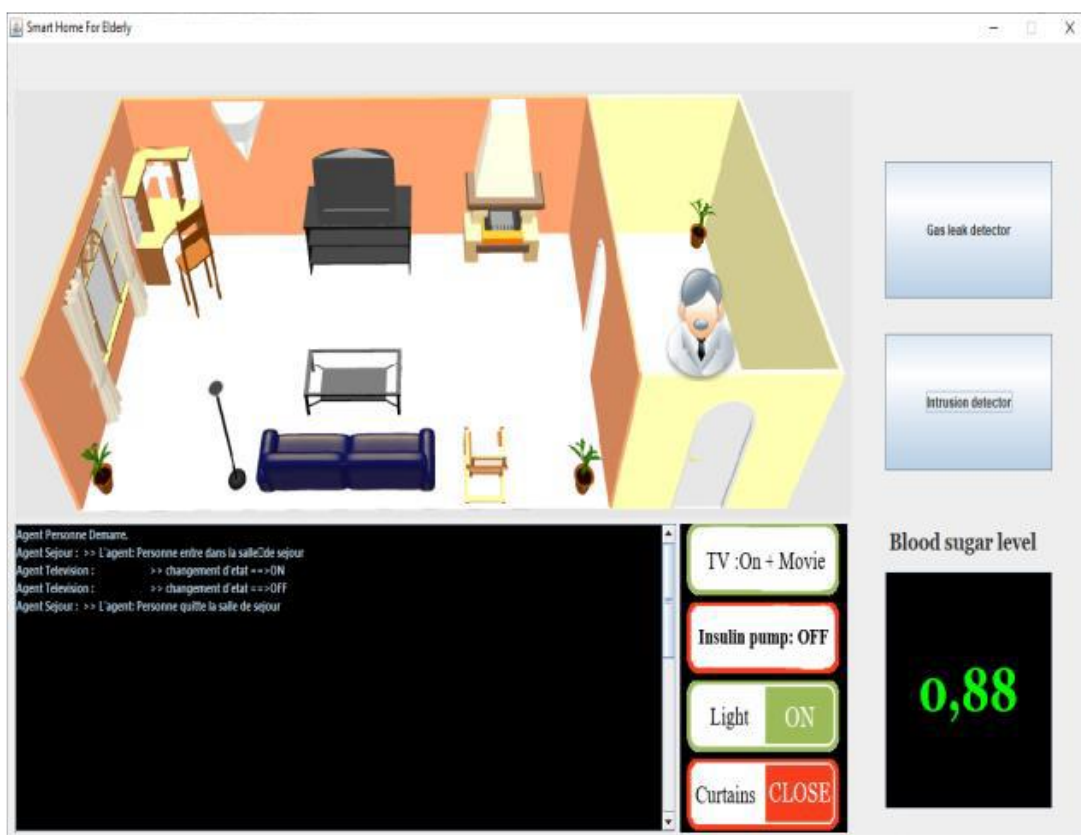


Figure 5.9 : Interface graphique de l'application de la pompe à insuline intelligente.

L'agent Sniffer est un agent JADE qui permet le suivi des échanges de messages entre les agents enregistrés dans JADE, permettant l'analyse de ces messages. La figure 5.10 montre la communication entre les agents de notre système, capturée par l'agent Sniffer.

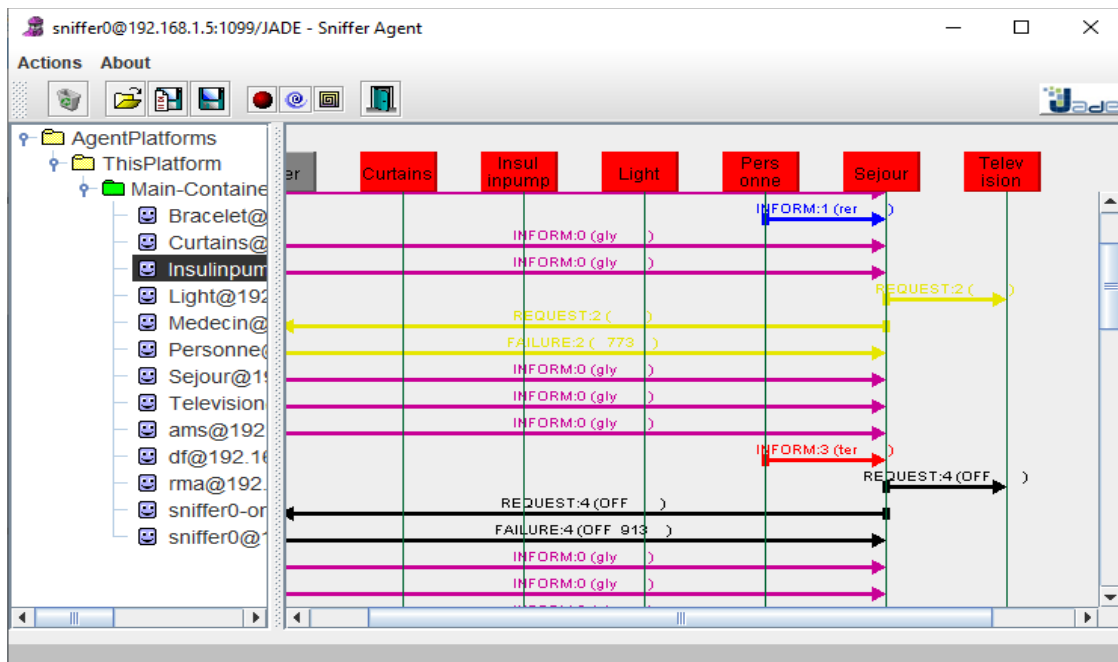


Figure 5.10 : L’outil Agent Sniffer capture les échanges de messages entre agents dans le système de la maison intelligente.

5.8 Discussion

L’approche proposée est axée sur l’utilisation individuelle dans un environnement de maison intelligente. La mise à l’échelle du système pour prendre en charge plusieurs utilisateurs peut présenter des défis, comme l’augmentation du volume de données et le besoin de mécanismes avancés de préservation de la vie privée. Cela pourrait être réalisé en explorant des solutions basées sur le cloud ou des architectures distribuées, qui accueilleraient plusieurs utilisateurs et s’intégreraient à de plus grands systèmes de santé intelligents. Cela comprend le développement de mécanismes pour l’agrégation et l’analyse efficaces des données dans plusieurs instances.

De plus, même si le système fonctionne bien dans des conditions normales, sa capacité à gérer les pics rapides de données — comme les multiples alertes déclenchées par des anomalies de capteurs ou des pannes de communication — pourrait avoir une incidence sur la prise de décision en temps réel. Des améliorations de la puissance de traitement ou des algorithmes optimisés peuvent être nécessaires pour maintenir les performances pendant les pics de charge.

Enfin, la robustesse du système en cas de dysfonctionnement des capteurs, de pannes de communication ou de bogues logiciels nécessite une évaluation plus approfondie. La mise en place de mécanismes de redondance ou de capacités d’auto-guérison pourrait

atténuer ces risques. La collaboration avec les fournisseurs de soins de santé pourrait également offrir des renseignements précieux pour des améliorations itératives.

Malgré son accent sur la prise en charge du diabète, le cadre a un potentiel important au-delà de ce cas d'utilisation spécifique. La souplesse et l'évolutivité du modèle ACOSO le rendent très applicable à d'autres problèmes de santé chroniques et aux applications des soins aux personnes âgées. En plus du diabète, le modèle ACOSO peut être utilisé pour gérer d'autres maladies chroniques, comme l'hypertension, les maladies cardiovasculaires et les affections respiratoires. Par exemple, des objets intelligents peuvent surveiller la pression artérielle, la fréquence cardiaque et les niveaux d'oxygène, déclenchant automatiquement des interventions ou alertant les soignants en cas de lectures irrégulières.

Dans des applications plus larges de soins aux personnes âgées, le modèle ACOSO pourrait être intégré à des systèmes de détection des chutes, d'observance des médicaments, d'interaction sociale et d'intervention d'urgence. La capacité des objets intelligents à communiquer et à répondre aux besoins de l'utilisateur, associée à l'adaptabilité du système multi-agents, pourrait améliorer la qualité de vie globale des personnes âgées en fournissant un suivi en temps réel, des soins personnalisés, et une sécurité accrue dans un cadre familial.

Le modèle ACOSO est également décentralisé, ce qui le rend très adaptable aux différents besoins des patients. Par exemple, des plans de soins personnalisés peuvent être mis en œuvre en fonction des données et préférences personnelles sur la santé, tandis que le système peut facilement évoluer pour intégrer d'autres appareils et services au fur et à mesure que les besoins du patient évoluent.

5.9 Conclusion

Assurer la santé et le bien-être des personnes âgées, en particulier celles souffrant de maladies chroniques multiples, demeure un objectif primordial dans le domaine des soins de santé. Parmi les maladies courantes touchant les personnes âgées, le diabète représente une préoccupation majeure, nécessitant des solutions innovantes pour améliorer leur qualité de vie. En réponse à ce besoin pressant, nous proposons dans ce chapitre une méthode centrée sur le développement d'une pompe à insuline intelligente. Ce dispositif surveille non seulement l'état de santé du patient de près, mais il est également capable de réagir rapidement en cas d'urgence, assurant ainsi une intervention médicale rapide.

L'approche présentée dans cette recherche s'appuie sur le modèle ABC, qui tire parti du potentiel du modèle ACOSO pour aborder efficacement les complexités de la gestion du

diabète chez les personnes âgées. En combinant ces modèles, nous établissons un cadre solide visant à alléger le fardeau de la surveillance constante de la santé pour les patients et leurs soignants.

Pour améliorer la précision et l'adaptabilité du système Smart InsulinPump proposé, des recherches futures pourraient explorer les algorithmes de prise de décision avancés. Il peut s'agir de modèles prédictifs pour la prévision des niveaux de glucose, de systèmes de contrôle adaptatif comme l'apprentissage par renforcement et de modèles personnalisés de glucose-insuline adaptés à la physiologie individuelle. Intégrer la prise de décisions contextuelles, la fusion de données multi-capteurs et l'optimisation liée à la sécurité pourraient améliorer encore la fiabilité et la sécurité des utilisateurs. Ces avancées visent à fournir des solutions d'administration de l'insuline plus précises, robustes et conviviales.

Conclusion Générale

&

Perspectives

Conclusion générale

Au cours de cette thèse, nous avons proposé une approche pour la modélisation et la simulation d'une maison intelligente pour les personnes âgées. Pour ce faire, nous avons utilisé l'outil Renew, qui est un simulateur très puissant et facile à utiliser pour modéliser des systèmes multi-agents.

À notre avis, le paradigme des réseaux dans les réseaux est le plus approprié pour de tels systèmes car il repose sur un formalisme dont la sémantique autorise donc formellement la vérification et l'exécution des systèmes mobiles.

L'utilisation du paradigme des réseaux dans les réseaux fournit une vue synthétique et permet un contrôle parfait du système dans son ensemble. En d'autres termes, il suffit de double-cliquer sur un jeton particulier pour que Renew ouvre le réseau schématisant le jeton choisi, permettant ainsi une inspection complète du système pendant la simulation sans aucune interruption. L'avantage est double : (1) utiliser les réseaux dans les réseaux comme paradigme de modélisation, conduit directement à un modèle exécutable du système. Pour atteindre un tel résultat, un processus de modélisation classique nécessitera au moins trois étapes : (a) la modélisation du système, (b) la mise en œuvre du modèle et (c) l'écriture du programme de visualisation, (2) visualiser et observer l'exécution du système modélisé n'est autre que la mise en œuvre du système lui-même. Cela élimine toute possibilité d'introduire des erreurs en passant du modèle à la visualisation des résultats via son implémentation. Nous avons augmenté l'outil de modélisation Renew par une connexion à un système expert, en l'occurrence Java Expert System Shell (JESS), ce qui constitue notre deuxième contribution.

Assurer la bonne santé et le bien-être des personnes âgées, en particulier celles qui souffrent de maladies chroniques multiples, reste un objectif primordial dans le domaine des soins de santé. Parmi les maladies courantes affectant les personnes âgées, le diabète se distingue comme une préoccupation majeure, nécessitant des solutions innovantes pour améliorer leur qualité de vie. En réponse à ce besoin pressant, La troisième approche présentée dans cette recherche introduit une méthode axée sur le développement d'une pompe à insuline intelligente. Cet appareil non seulement surveille de près l'état de santé du patient, mais a également la capacité de réagir rapidement en cas d'urgence, assurant ainsi une intervention médicale rapide.

Cette approche s'appuie sur le modèle ABC, tirant parti du potentiel du modèle ACOSO pour aborder efficacement les complexités de la gestion du diabète chez la population âgée.

Conclusion générale

L'architecture ACOSO (Agent-based COoperating Smart Objects) présente un avantage majeur en favorisant la coopération intelligente et l'autonomie des objets connectés au sein d'un environnement distribué. En s'appuyant sur le paradigme des systèmes multi-agents, chaque objet intelligent est capable de percevoir son environnement, de prendre des décisions locales et de collaborer avec d'autres objets pour atteindre des objectifs communs. Cette approche se distingue par sa flexibilité, permettant une intégration aisée de nouveaux dispositifs, ainsi que par son évolutivité, qui facilite l'adaptation du système à des contextes variés. De plus, ACOSO contribue à une optimisation des ressources grâce à une coordination efficace entre agents, tout en assurant une robustesse accrue face aux défaillances partielles du réseau. Enfin, la coopération entre objets intelligents engendre une forme d'intelligence collective, essentielle pour la gestion efficace et autonome des environnements complexes de l'Internet des objets.

Perspectives

Dans les travaux futurs, nous avons l'intention d'améliorer notre approche en incluant les autres agents de la maison intelligente afin qu'ils collaborent et coopèrent pour le bien-être des habitants de cette maison. Nous prendrons en considération la prévention des incendies, les accidents auxquels seront confrontées les personnes.

Pour renforcer la précision et l'adaptabilité du système de la pompe d'insuline intelligente proposé, plusieurs axes de recherche peuvent être envisagés. Des algorithmes avancés de prise de décision pourraient être explorés, notamment des modèles prédictifs pour l'anticipation du niveau de glucose, des systèmes de contrôle adaptatifs basés sur l'apprentissage par renforcement, ainsi que des modèles personnalisés glucose-insuline tenant compte des caractéristiques physiologiques individuelles. L'intégration d'une prise de décision contextuelle, la fusion de données multi-capteurs et l'optimisation sous contraintes de sécurité constituent également des pistes prometteuses pour améliorer la robustesse et la sécurité du dispositif. Ces avancées permettraient de proposer des solutions d'administration d'insuline plus précises, fiables et conviviales.

Par ailleurs, les perspectives futures pourraient s'étendre au-delà de la seule gestion du diabète, pour englober un ensemble plus large de services dédiés aux soins des personnes âgées. En privilégiant la sécurité et le confort dans leur environnement domestique, l'objectif serait de renforcer le concept de vie autonome pour les personnes âgées susceptibles d'être orientées vers des institutions spécialisées. La vision globale vise à concevoir un environnement intelligent

Conclusion générale

qui soutienne la santé, tout en favorisant l'autonomie et la dignité. Dans cette optique, l'intégration de technologies avancées, d'interfaces intuitives et de plans de soins personnalisés pourrait constituer un levier essentiel pour améliorer durablement leur qualité de vie.

Bibliographie

- [1] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggle, P. (1999). Towards a Better Understanding of Context and Context-Awareness. *Ubiquitous Computing*, 304-307. <https://doi.org/10.5555/647985.743843>.
- [2] Abras, S. (2009). Système domotique multi-agents pour la gestion de l'énergie dans l'habitat. <https://www.theses.fr/2009INPG0049>.
- [3] Acampora, G., Cook, D. J., Rashidi, P., & Vasilakos, A. V. (2013). A Survey on Ambient Intelligence in Healthcare. *Proceedings Of The IEEE*, 101(12), 2470-2494. <https://doi.org/10.1109/jproc.2013.2262913>.
- [4] Adeniyi, A. E., Awotunde, J. B., Falola, P. B., & Aworinde, H. O. (2024). Application of Blockchain-Based Internet of Things in Medical Healthcare. Dans *Advances in civil and industrial engineering book series* (p.233-266). <https://doi.org/10.4018/979-8-3693-8568-5.ch010>.
- [5] Agarwal, S., Crespo-Ramos, G., Long, J. A., & Miller, V. A. (2021). "I Didn't Really Have a Choice": Qualitative Analysis of Racial-Ethnic Disparities in Diabetes Technology Use Among Young Adults with Type 1 Diabetes. *Diabetes Technology & Therapeutics*, 23(9), 616-622. <https://doi.org/10.1089/dia.2021.0075>.
- [6] Agoulmine, N., Deen, M., Lee, J., & Meyyappan, M. (2011). U-Health Smart Home. *IEEE Nanotechnology Magazine*, 5(3), 6-11. <https://doi.org/10.1109/mnano.2011.941951>.
- [7] Aipperspach, R., Cohen, E., & Canny, J. (2006). Modeling Human Behavior from Simple Sensors in the Home. Dans *Lecture notes in computer science* (p. 337-348). https://doi.org/10.1007/11748625_21.
- [8] Akyildiz, I., Su, W., Sankarasubramanian, Y., & Cayirci, E. (2002). Wireless sensor networks : a survey. *Computer Networks*, 38(4), 393-422. [https://doi.org/10.1016/s1389-1286\(01\)00302-4](https://doi.org/10.1016/s1389-1286(01)00302-4).
- [9] Alshamrani, M. (2021). IoT and artificial intelligence implementations for remote healthcare monitoring systems : A survey. *Journal Of King Saud University - Computer And Information Sciences*, 34(8), 4687-4701. <https://doi.org/10.1016/j.jksuci.2021.06.005>.

Bibliographie

- [10] Alti, A., & Laouamer, L. (2021). Agent-Based Autonomic Semantic Context-Aware Platform for Smart Health Monitoring and Disease Detection. *The Computer Journal*, 65(3), 736-755. <https://doi.org/10.1093/comjnl/bxab075>.
- [11] Baldauf, M., Dustdar, S., & Rosenberg, F. (2007b). A survey on context-aware systems. *International Journal Of Ad Hoc And Ubiquitous Computing*, 2(4), 263. <https://doi.org/10.1504/ijahuc.2007.014070>.
- [12] Bellifemine, F., Poggi, A., & Rimassa, G. (2007). *Developing Multi-Agent Systems with JADE*. <https://doi.org/10.1002/9780470058411>.
- [13] Benazzouz, Y. (2011). Découverte de contexte pour une adaptation automatique de services en intelligence ambiante. <http://www.theses.fr/2011EMSE0617.pdf>.
- [14] Benazzouz, Y. (2012). Découverte de connaissances en intelligence ambiante. Dans Presses Académiques Francophones eBooks. <https://www.knigozal.com/store/gb/book/d%C3%A9couverte-de-connaissances-en-intelligence-ambiante/isbn/978-3-8381-8821-8>.
- [15] Blair, J., McKay, A., Ridyard, C., Thornborough, K., Bedson, E., Peak, M., Didi, M., Annan, F., Gregory, J. W., Hughes, D., & Gamble, C. (2018). Continuous subcutaneous insulin infusion versus multiple daily injections in children and young people at diagnosis of type 1 diabetes: the SCIPI RCT. *Health Technology Assessment*, 22(42), 1-112. <https://doi.org/10.3310/hta22420>.
- [16] Bouaita, A., Kissoum, Y., & Mazouzi, S. (2022, December). The Multi Agent Nets Architecture as Base for Modelling Smart Home for Elderly: International Symposium on iNnovative Informatics of Biskra (ISNIB),1-6. <https://doi.org/10.1109/isnib57382.2022.10076104>.
- [17] Bouaita, A., Kissoum, Y., & Mazouzi, S. (2024). Agent based smart insulin pump for autonomous diabetes management. *International Journal Of Simulation And Process Modelling*, 21(4), 238-250. <https://doi.org/10.1504/ijspm.2024.146022>.
- [18] Brown, Peter J. (1996) *The Stick-e Document: a Framework for Creating Context-aware Applications*. Electronic Publishing, 8:259–272. <https://kar.kent.ac.uk/21408/>.

Bibliographie

- [19] Castelfranchi, C., & Lespérance, Y. (2001). Intelligent Agents VII Agent Theories Architectures and Languages. Dans *Lecture notes in computer science*. <https://doi.org/10.1007/3-540-44631-1>.
- [20] Cazañas-Gordón, A., & Parra-Mora, E. (2020). The Internet of Things in Healthcare. An Overview. *Zenodo (CERN European Organization For Nuclear Research)*. <https://doi.org/10.5281/zenodo.5730386>.
- [21] Chaari, T. (2007). *Adaptation d'applications pervasives dans des environnements multi-contextes*. <http://docinsa-breves.insa-lyon.fr/these/2007/chaari/these.pdf>.
- [22] Chang, X., Huang, P., Shen, Y., Liang, X., Yang, Y., & Hauptmann, A. G. (2018b). RCAA : Relational Context-Aware Agents for Person Search. Dans *Lecture notes in computer science* (p. 86-102). https://doi.org/10.1007/978-3-030-01240-3_6.
- [23] Clarizia, F., Lemma, S., Lombardi, M., & Pascale, F. (2017). A Mobile Context-Aware Information System to Support Tourism Events. Dans *Lecture notes in computer science* (p. 553-566). https://doi.org/10.1007/978-3-319-57186-7_40.
- [24] Cook, D. J. (2009). Multi-agent smart environments. *Journal Of Ambient Intelligence And Smart Environments*, 1(1), 51-55. <https://doi.org/10.3233/ais-2009-0007>.
- [25] D'Inverno, M., & Luck, M. (2004). Understanding agent systems. Dans *Springer series in agent technology/Springer series on agent technology*. <https://doi.org/10.1007/978-3-662-10702-7>.
- [26] Dorri, A., Kanhere, S. S., & Jurdak, R. (2018). Multi-Agent Systems : a survey. *IEEE Access*, 6, 28573-28593. <https://doi.org/10.1109/access.2018.2831228>.
- [27] Fautrero, V., Fernandez, V., & Khalil, S. (2018). Le paradigme du cloud computing : au-delà de nouvelles solutions informatiques, un enjeu de gouvernance renouvelée des technologies numériques. *Annales des Mines - Gérer et Comprendre*, N° 133(3), 13-24. <https://doi.org/10.3917/geco1.133.0013>.
- [28] Fortino, G., Giannantonio, R., Gravina, R., Kuryloski, P., & Jafari, R. (2012). Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications. *IEEE Transactions On Human-Machine Systems*, 43(1), 115-133. <https://doi.org/10.1109/tsmcc.2012.2215852>.

- [29] Fortino, G., Guerrieri, A., O'Hare, G., & Ruzzelli, A. (2012). A flexible building management framework based on wireless sensor and actuator networks. *Journal Of Network And Computer Applications*, 35(6), 1934-1952. <https://doi.org/10.1016/j.jnca.2012.07.016>.
- [30] Fortino, G., Guerrieri, A., Russo, W., & Savaglio, C. (2014, May). Integration of agent-based and cloud computing for the smart objects-oriented IoT. In *Proceedings of the 2014 IEEE 18th international conference on computer supported cooperative work in design (CSCWD)*, 493-498. <https://doi.org/10.1109/cscwd.2014.6846894>.
- [31] Fortino, G., Guerrieri, A., Russo, W., & Savaglio, C. (2014). Middlewares for Smart Objects and Smart Environments : Overview and Comparison. Dans *Internet of things* (p. 1-27). https://doi.org/10.1007/978-3-319-00491-4_1.
- [32] Fortino, G., Lackovic, M., Russo, W., & Trunfio, P. (2013). A discovery service for smart objects over an agent-based middleware. In *Internet and Distributed Computing Systems: 6th International Conference, IDCS 2013, Hangzhou, China, Proceedings 6*, 281-293. https://doi.org/10.1007/978-3-642-41428-2_23.
- [33] Fortino, G., Russo, W., Savaglio, C., Shen, W., & Zhou, M. (2017). Agent-Oriented Cooperative Smart Objects: From IoT System Design to Implementation. *IEEE Transactions On Systems Man And Cybernetics Systems*, 48(11), 1939-1956. <https://doi.org/10.1109/tsmc.2017.2780618>.
- [34] Gehlot, V., & Nigro, C. (2010). An introduction to systems modeling and simulation with Colored Petri Nets. *Proceedings Of Winter Simulation Conference*, 104-118. <https://doi.org/10.1109/wsc.2010.5679170>.
- [35] Gulati, K., Boddu, R. S. K., Kapila, D., Bangare, S. L., Chandnani, N., & Saravanan, G. (2021). A review paper on wireless sensor network techniques in Internet of Things (IoT). *Materials Today Proceedings*, 51, 161-165. <https://doi.org/10.1016/j.matpr.2021.05.067>.
- [36] Hartmann, M. (2010). *Context-Aware Intelligent User Interfaces for Supporting System Use* (p. 1-185). <https://tuprints.ulb.tu-darmstadt.de/2165/>.
- [37] Hernandez, N. (2005). *Ontologies de domaine pour la modélisation du contexte en recherche d'Information*. <https://tel.archives-ouvertes.fr/tel-00371323>.
- [38] Jensen, K. (1997b). A brief introduction to coloured Petri Nets. Dans *Lecture notes in computer science* (p. 203-208). <https://doi.org/10.1007/bfb0035389>.

- [39] Jensen, K., & Kristensen, L. M. (2015). Colored Petri nets. *Communications Of The ACM*, 58(6), 61-70. <https://doi.org/10.1145/2663340>.
- [40] Karlson, B., Bria, A., Lind, J., Lönnqvist, P., & Norlin, C. (2003). *Wireless Foresight : Scenarios of the Mobile World in 2015*. <http://doi.wiley.com/10.1002/0470858265>.
- [41] Kaushik, S., & Tyagi, K. (2016b). Critical Review on Test Case Generation Systems and Techniques. *International Journal Of Computer Applications*, 133(7), 24-29. <https://doi.org/10.5120/ijca2016907916>.
- [42] Köhler, M., Moldt, D., & Rölke, H. (2001). Modelling the Structure and Behaviour of Petri Net Agents. Dans *Lecture notes in computer science* (p. 224-241). https://doi.org/10.1007/3-540-45740-2_14.
- [43] Köhler, M., Moldt, D., & Rölke, H. (2003c). Modelling Mobility and Mobile Agents Using Nets within Nets. Dans *Lecture notes in computer science* (p. 121-139). https://doi.org/10.1007/3-540-44919-1_11.
- [44] Kummer O, Wienberg F and Duvigneau M(2002), *Renew – User Guide*, University of Hamburg, Computer Science Department, Vogt-Kölln Str. 30, 22527 Hamburg, Deutschland, 1.6 edition.
- [45] Lakos, C. A. (2005). A Petri Net View of Mobility. Dans *Lecture notes in computer science* (p. 174-188). https://doi.org/10.1007/11562436_14.
- [46] Liu, H., Gegov, A., & Stahl, F. (2014). Categorization and Construction of Rule Based Systems. In *Communications in computer and information science* (p. 183 194). https://doi.org/10.1007/978-3-319-11071-4_18.
- [47] Li, H., Guo, Y., Zhao, H., Wang, Y., & Chow, D. (2021). Towards automated greenhouse: A state-of-the-art review on greenhouse monitoring methods and technologies based on internet of things. *Computers And Electronics In Agriculture*, 191, 106558. <https://doi.org/10.1016/j.compag.2021.106558>.
- [48] Luck, M., McBurney, P., & Preist, C. (2004). A Manifesto for Agent Technology: Towards Next Generation Computing. *Autonomous Agents And Multi-Agent Systems*, 9(3), 203 252. <https://doi.org/10.1023/b:agnt.0000038027.29035.7c>.
- [49] Machorro-Cano, I., Olmedo-Aguirre, J. O., Alor-Hernández, G., Rodríguez-Mazahua, L., Sánchez-Morales, L. N., & Pérez-Castro, N. (2023). Cloud-Based Platforms for Health Monitoring: A Review. *Informatics*, 11(1), 2. <https://doi.org/10.3390/informatics1101000>.

Bibliographie

- [50] Maes, P. (1994). Agents that reduce work and information overload. *Communications Of The ACM*, 37(7), 30-40. <https://doi.org/10.1145/176789.176792>.
- [51] Maraqa, O., Rajasekaran, A. S., Al-Ahmadi, S., Yanikomeroglu, H., & Sait, S. M. (2020). A Survey of Rate-Optimal Power Domain NOMA With Enabling Technologies of Future Wireless Networks. *IEEE Communications Surveys & Tutorials*, 22(4), 2192-2235. <https://doi.org/10.1109/comst.2020.3013514>.
- [52] Martin, C., Fiorino, H., & Calvary, G. (2015). Planification flexible. Un besoin en intelligence ambiante. Un défi en planification automatique. *Revue D Intelligence Artificielle*, 29(1), 11-46. <https://doi.org/10.3166/ria.29.11-46>.
- [53] Moldt, D. (2000). Workshop on Modelling of Objects, Components, and Agents, Aarhus, Denmark, August 27-28, 2001. *DAIMI Report Series*, 30(553). <https://doi.org/10.7146/dpb.v30i553.7110>.
- [54] Moldt, D. (2004b). Third Workshop on Modelling of Objects, Components, and Agents. *DAIMI Report Series*, 33(571). <https://doi.org/10.7146/dpb.v33i571.7128>.
- [55] Morris, M. E., Adair, B., Miller, K., Ozanne, E., Hansen, R., Pearce, A. J., Santamaria, N., Viegas, L., Long, M., & Said, C. M. (2013). Smart-Home Technologies to Assist Older People to Live Well at Home. *Journal Of Aging Science*, 01(01). <https://doi.org/10.4172/2329-8847.1000101>.
- [56] Naïditch, N., & Riveline, J. (2023). Diabétologie connectée : quelles sont les attentes des médecins et des patients ? *Médecine des Maladies Métaboliques*, 17(2), 2S3-2S10. [https://doi.org/10.1016/s1957-2557\(23\)00073-1](https://doi.org/10.1016/s1957-2557(23)00073-1).
- [57] Nguyen, D. C., Ding, M., Pathirana, P. N., Seneviratne, A., Li, J., & Poor, H. V. (2021). Federated Learning for Internet of Things : A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1622-1658. <https://doi.org/10.1109/comst.2021.3075439>.
- [58] Noury, N., Virone, G., Barralon, P., Rialle, V., & Demongeot, J. (2004). Maisons intelligentes pour personnes âgées : technologies de l'information intégrées au service des soins à domicile. *J3eA*, 3, 020. <https://doi.org/10.1051/bib-j3ea:2004620>.
- [59] Pascoe, J. (2002). Adding generic contextual capabilities to wearable computers. *Second International Symposium On Wearable Computers*. <https://doi.org/10.1109/iswc.1998.729534>.

- [60] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2013). Context Aware Computing for The Internet of Things : A Survey. *IEEE Communications Surveys & Tutorials*, 16(1), 414-454. <https://doi.org/10.1109/surv.2013.042313.00197>.
- [61] Poslad, S. (2009). *Ubiquitous Computing : Smart Devices, Environments and Interactions*. <http://ci.nii.ac.jp/ncid/BA90648752>.
- [62] Rancea, A., Anghel, I., & Cioara, T. (2024). Edge Computing in Healthcare: Innovations, Opportunities, and Challenges. *Future Internet*, 16(9), 329. <https://doi.org/10.3390/fi16090329>.
- [63] Reignier, P. (2012). Intelligence Ambiante ProActive. Dans *Editions universitaires europeennes eBooks*. <https://www.knigozal.com/store/gb/book/intelligence-ambiante-proactive/isbn/978-3-8417-9856-5>.
- [64] Renew the reference networkshop.WWW page at <http://renew.de/>. Contains the documentation for Renew and an introduction to reference nets.
- [65] Sahraoui, S. (2016b). *Mécanismes de sécurité pour l'intégration des RCSFs à l'IoT (Internet of Things)*. <http://eprints.univ-batna2.dz/308/>.
- [66] Saleh, I. (2018b). Internet des objets (IDO) : concepts, enjeux, défis et perspectives. *Internet des Objets*, 2(1). <https://doi.org/10.21494/iste.op.2018.0229>.
- [67] Savaglio, C., Fortino, G., Ganzha, M., Paprzycki, M., Bădică, C., & Ivanović, M. (2017). Agent-Based Computing in the Internet of Things : A Survey. Dans *Studies in computational intelligence* (p. 307-320). https://doi.org/10.1007/978-3-319-66379-1_27.
- [68] Schilit, B., Adams, N., & Want, R. (1994). Context-Aware Computing Applications. *First Workshop On Mobile Computing Systems And Applications*, 85-90. <https://doi.org/10.1109/wmcsa.1994.16>.
- [69] Schmidt, K., & Stahl, C. (2005). 12. Workshop « Algorithmen und Werkzeuge für Petrinetze » (AWPN 2005). Dans *Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, Institut für Informatik eBooks*. <https://doi.org/10.18452/2454>.

Bibliographie

- [70] Staszewski, W. J., Boller, C., & Tomlinson, G. R. (2017). *Health monitoring of aerospace structures : smart sensor technologies and signal processing*. <http://ci.nii.ac.jp/ncid/BA66790145>.
- [71] Siebers, P., & Klügl, F. (2017). What Software Engineering Has to Offer to Agent-Based Social Simulation. Dans *Understanding complex systems* (p. 81-117). https://doi.org/10.1007/978-3-319-66948-9_6.
- [72] Silva, M., & Recalde, L. (2002b). Petri nets and integrality relaxations : a view of continuous petri net models. *IEEE Transactions On Systems Man And Cybernetics Part C (Applications And Reviews)*, 32(4), 314-327. <https://doi.org/10.1109/tsmcc.2002.806063>.
- [73] Sovacool, B. K., Del Rio, D. D. F., Bergman, N., & Makuch, K. E. (2020). Critically reviewing smart home technology applications and business models in Europe. *Energy Policy*, 144, 111631. <https://doi.org/10.1016/j.enpol.2020.111631>.
- [74] Tapia, E. M., Intille, S. S., & Larson, K. (2004b). Activity Recognition in the Home Using Simple and Ubiquitous Sensors. Dans *Lecture notes in computer science* (p. 158-175). https://doi.org/10.1007/978-3-540-24646-6_10.
- [75] Touati, Y. E., Alouane, N. B. H., & Yeddes, M. (2005b). Réseau de Petri temporel étendu. Proposition d'un nouveau modèle basé sur les réseaux de Petri temporels. *Journal Européen des Systèmes Automatisés*, 39(1-3), 207-222. <https://doi.org/10.3166/jesa.39.207-222>.
- [76] Valette, R. (2009). Introduction aux Réseaux de Petri. *Bull. d'Informatique Approfondie et Appl.*, 83, 3-22. <http://dblp.uni-trier.de/db/journals/biaa/biaa83.html#Valette09a>.
- [77] Van Dam, S. S., Bakker, C. A., & Van Hal, J. D. M. (2010). Home energy monitors : impact over the medium-term. *Building Research & Information*, 38(5), 458-469. <https://doi.org/10.1080/09613218.2010.494832>.
- [78] Van Hee, K. M., Lomazova, I. A., Oanea, O., Serebrenik, A., Sidorova, N., & Voorhoeve, M. (2006). Nested Nets for Adaptive Systems. Dans *Lecture notes in computer science* (p. 241-260). https://doi.org/10.1007/11767589_14.
- [79] Vuillemin, B. (2020b). *Recherche de règles de prédiction dans un contexte d'Intelligence Ambiante*. <http://www.theses.fr/2020LYSE1120>.

Bibliographie

- [80] Xu, N. D., & Deng, N. Y. (2002). Modeling mobile agent systems with high level Petri nets. *The IEEE International Conference On Systems, Man, And Cybernetics (SMC'00)*, 5, 3177-3182. <https://doi.org/10.1109/icsmc.2000.886486>.
- [81] Yabe, D., & Seino, Y. (2014). Dipeptidyl peptidase-4 inhibitors and sulfonylureas for type 2 diabetes: Friend or foe? *Journal Of Diabetes Investigation*, 5(5), 475-477. <https://doi.org/10.1111/jdi.12229>.
- [82] Yacine, K., Ramdane, M., & Zaidi, S. (2014). Modeling a Smart Home for Elderly. In *Model and Data Engineering: 4th International Conference, MEDI 2014, Larnaca, Cyprus*, 313-320. https://doi.org/10.1007/978-3-319-11587-0_29.
- [83] Zhu, J., Shi, K., Yang, C., Niu, Y., Zeng, Y., Zhang, N., Liu, T., & Chu, C. H. (2021). Ethical issues of smart home-based elderly care: A scoping review. *Journal Of Nursing Management*, 30(8), 3686-3699. <https://doi.org/10.1111/jonm.13>.