

**UN HABITAT CAPABLE DE RECONNAÎTRE LES  
ACTIVITÉS PLANIFIÉES DANS UN CALENDRIER  
ÉLECTRONIQUE**

par

Pierre-Yves Nivollet

Mémoire présenté au Département d'informatique  
en vue de l'obtention du grade de maître ès sciences (M.Sc.)

**FACULTÉ DES SCIENCES  
UNIVERSITÉ DE SHERBROOKE**

Sherbrooke, Québec, Canada, 29 novembre 2013

# Sommaire

Le vieillissement de la population est un problème qui touche de plus en plus le monde occidental et a pour conséquence de rendre difficile le maintien à domicile des personnes âgées, étant donné les coûts engendrés et le manque de personnel soignant. D'un autre côté, la présence des technologies dans les domiciles devient répandue et ce, à des prix abordables. On est désormais capable de récupérer facilement des informations sur ce qui se passe à l'intérieur de la maison et ainsi agir en conséquence. Le maintien à domicile devient alors possible en offrant une assistance aux personnes âgées qui soit portée par la technologie, afin de les aider à réaliser leurs activités de la vie quotidienne. À l'aide d'un calendrier électronique dans lequel seront notées toutes les activités à réaliser dans la journée, on est en mesure d'utiliser les données fournies par l'appartement pour vérifier qu'elles ont bien été réalisées, et ainsi offrir des rappels efficaces pour ces activités.

C'est vers cette problématique que se tourne ce mémoire en offrant un outil pouvant utiliser et interpréter les informations provenant de l'intérieur d'un habitat, combinées à celles venant d'un calendrier électronique, dans le but d'aider à la réalisation des tâches de la vie quotidienne. Cet outil est composé de deux programmes principaux : le calendrier électronique et le programme de reconnaissance d'activité, les deux fonctionnant en collaboration.

Ce mémoire présente en premier lieu l'informatique ubiquitaire, les moyens de rappel existants ainsi que les différentes méthodes pour faire de la reconnaissance d'activité dans un appartement intelligent. Suit une description du calendrier électronique permettant aux personnes âgées d'entrer correctement leurs activités à rappeler. Ensuite est décrit le programme de reconnaissance d'activité, qui interprète les événements des capteurs pour reconnaître les activités réalisées. Le chapitre suivant traite

## SOMMAIRE

de la collaboration entre le calendrier et le programme de reconnaissance d'activité afin qu'ils soient utilisables dans l'appartement. Une expérimentation réalisée dans l'appartement intelligent du DOMUS cherche à fournir une preuve de concept du rappel d'activité en fonction de ce qui se passe dans l'appartement.

**Mots-clés:** habitat intelligent ; informatique ubiquitaire ; agenda électronique ; personnes âgées atteintes de démence ; maintien à domicile ; reconnaissance d'activité ; mixture de Gaussiennes ; algorithme EM ; K-Moyenne.

# Remerciements

Je tiens premièrement à remercier ma directrice de recherche H el ene Pigot, co-directrice du laboratoire DOMUS, de m'avoir offert l'opportunit e de faire ce travail. Je souhaite  galement la remercier pour son suivi rigoureux, sa disponibilit e et les nombreux conseils qu'elle a su me donner.

Je d sire aussi remercier toute l' quipe du laboratoire DOMUS, en particulier Jean-Paul Viboud qui a su me guider dans la programmation et l'int gration d'AMELIS. Je remercie aussi Yannick Adeline pour ses nombreux conseils et le temps pass e avec moi pour m'aider   bien cibler mon sujet. Je remercie  galement Sylvain Giroux pour sa bonne humeur et la pertinence de ses interrogations concernant mon travail. Je tiens aussi   remercier tous les  tudiants et stagiaires qui ont su mettre de la vie dans ce laboratoire. Cette ma trise n'aurait pas  t  pareille sans la bonne ambiance et la joie de vivre qui y r gnaient.

Finalement, j'adresse mes remerciements   mes parents pour leur soutien et leur aide apport e tout au long de cette ma trise, malgr e la distance nous s parant.

# Abréviations

**AVQ** Activité de la Vie Quotidienne

**AMELIS** Alzheimer Mémoire et Liens Sociaux

**PRADHA** Programme de Reconnaissance d'Activité Dans l'Habitat

**RFID** *Radio Frequency IDentification*

**MMC** Modèle de Markov Caché

**CBN** Classificateur Bayésien Naïf

**SGBD** Système de Gestion de Base de données

**BD** Base de Données

**MVC** Modèle-Vue-Contrôleur

**EM** Espérance-Maximisation

**VP** Vrai Positif

**VN** Vrai Négatif

**FP** Faux Positif

**FN** Faux Négatif

# Table des matières

<b>Sommaire</b>	<b>i</b>
<b>Remerciements</b>	<b>iii</b>
<b>Abréviations</b>	<b>iv</b>
<b>Table des matières</b>	<b>v</b>
<b>Liste des figures</b>	<b>ix</b>
<b>Liste des tableaux</b>	<b>xi</b>
<b>Introduction</b>	<b>1</b>
<b>1 État de l’art</b>	<b>5</b>
1.1 Un besoin pour les personnes âgées . . . . .	5
1.1.1 Une population vieillissante . . . . .	5
1.1.2 Le maintien à domicile . . . . .	6
1.1.3 Informatique ubiquitaire . . . . .	7
1.2 Les moyens de rappel existants dans un environnement intelligent . .	9
1.3 Reconnaissance d’activité dans un habitat . . . . .	14
1.3.1 Reconnaissance à partir de caméras . . . . .	14
1.3.2 Reconnaissance à partir de données des capteurs . . . . .	15
1.4 Bilan . . . . .	22

## TABLE DES MATIÈRES

<b>2</b>	<b>Objectifs et méthodologie</b>	<b>24</b>
2.1	Objectifs liés à AMELIS . . . . .	24
2.1.1	Objectifs au niveau des fonctionnalités . . . . .	24
2.1.2	Objectifs au niveau de la programmation . . . . .	25
2.2	Objectifs liés à PRADHA . . . . .	25
2.2.1	Objectifs au niveau de la programmation . . . . .	25
2.2.2	Objectifs au niveau de la reconnaissance . . . . .	26
2.2.3	Objectifs au niveau de l'utilisation d'AMELIS . . . . .	27
2.3	Méthodologie . . . . .	27
2.3.1	Réalisation du calendrier AMELIS . . . . .	27
2.3.2	Réalisation de PRADHA . . . . .	28
2.3.3	Validation de AMELIS-PRADHA . . . . .	29
2.4	Description de l'appartement du laboratoire . . . . .	30
2.4.1	Présentation . . . . .	30
2.4.2	Équipement . . . . .	31
<b>3</b>	<b>Projet AMELIS</b>	<b>34</b>
3.1	Introduction . . . . .	34
3.1.1	Présentation . . . . .	34
3.1.2	Interface . . . . .	35
3.1.3	Fonctionnalités . . . . .	36
3.1.4	Équipement . . . . .	38
3.2	Version partant de la base de données . . . . .	38
3.3	Version partant du modèle Java . . . . .	40
3.3.1	Architecture et fonctionnalités . . . . .	41
3.3.2	Représentation d'une activité . . . . .	42
3.3.3	Interface d'AMELIS . . . . .	43
3.4	Bilan . . . . .	43
<b>4</b>	<b>Projet Pradha</b>	<b>45</b>
4.1	Introduction . . . . .	45
4.1.1	Objectifs du projet PRADHA . . . . .	45
4.1.2	Jeu de données . . . . .	46

## TABLE DES MATIÈRES

4.2	Algorithme de simple association . . . . .	48
4.2.1	Apprentissage . . . . .	48
4.2.2	Validation . . . . .	49
4.2.3	Limites . . . . .	50
4.3	Algorithme de mixture de Gaussiennes . . . . .	51
4.3.1	Présentation générale . . . . .	51
4.3.2	Représentation des données . . . . .	52
4.3.3	Apprentissage . . . . .	53
4.3.4	Validation . . . . .	58
4.4	Bilan . . . . .	60
<b>5</b>	<b>Intégration d'AMELIS</b>	<b>64</b>
5.1	Présentation . . . . .	64
5.2	PRADHA dans l'appartement . . . . .	66
5.2.1	Récupération des évènements des capteurs . . . . .	66
5.2.2	Fonctionnement de la reconnaissance . . . . .	66
5.3	Communication entre AMELIS et PRADHA . . . . .	69
5.3.1	Patron observateur . . . . .	69
5.3.2	Gestion des rappels dans PRADHA . . . . .	70
5.4	Réponse d'AMELIS à l'utilisateur . . . . .	71
5.5	Bilan . . . . .	73
<b>6</b>	<b>Protocole de l'expérimentation</b>	<b>75</b>
6.1	Buts de l'expérimentation . . . . .	75
6.2	Expérimentation . . . . .	76
6.2.1	Population . . . . .	76
6.2.2	Déroulement . . . . .	76
<b>7</b>	<b>Résultats de l'expérimentation</b>	<b>81</b>
7.1	Résultats de la reconnaissance . . . . .	81
7.1.1	Données collectées . . . . .	81
7.1.2	Résultats de l'algorithme complet . . . . .	83
7.1.3	Résultats de l'algorithme de simple association . . . . .	85



## TABLE DES MATIÈRES

7.1.4	Résultats de l'algorithme de mixture de Gaussiennes . . . . .	88
7.1.5	Bilan . . . . .	90
7.2	Résultats d'utilisabilité du calendrier . . . . .	90
7.2.1	Analyse des résultats . . . . .	90
7.2.2	Bilan . . . . .	94
	<b>Conclusion</b>	<b>95</b>
	<b>A Diagramme de classe d'AMELIS</b>	<b>100</b>
	<b>B Questionnaire utilisé pour l'expérimentation</b>	<b>103</b>
B.1	Informations sur le participant . . . . .	103
B.2	Habitudes de vie . . . . .	103
B.3	Utilisation de PRADHA . . . . .	104
B.4	Appréciation de PRADHA . . . . .	105
	<b>C Résultats des expérimentations</b>	<b>106</b>

# Liste des figures

2.1	Plan du laboratoire DOMUS . . . . .	31
3.1	Capture de l'écran d'accueil d'AMELIS . . . . .	35
3.2	Capture de l'écran de détail de la journée . . . . .	36
3.3	Arbre des dépendances dans AMELIS . . . . .	41
3.4	Représentation des entités principales d'AMELIS . . . . .	42
4.1	Exemple d'un fichier d'annotations . . . . .	47
4.2	Exemple d'un fichier d'évènements de capteurs . . . . .	48
4.3	Fonctionnement de l'algorithme EM . . . . .	59
4.4	Résultats de reconnaissance des deux algorithmes . . . . .	61
5.1	Interactions entre AMELIS, PRADHA et l'utilisateur . . . . .	65
5.2	Reconnaissance d'activité dans l'appartement . . . . .	68
5.3	Lancement de la reconnaissance . . . . .	70
5.4	Classes permettant de faire de la reconnaissance d'activité . . . . .	71
5.5	Interface d'AMELIS pour effectuer le rappel . . . . .	73
5.6	Arbre des dépendances d'AMELIS et de PRADHA . . . . .	74
7.1	Précision de la reconnaissance par activité . . . . .	85
7.2	Résultats de l'algorithme de simple association . . . . .	86
7.3	Résultats de l'algorithme de mixture de Gaussiennes . . . . .	88
7.4	Nombre d'identifications de réalisation des activités par PRADHA . . . . .	92
A.1	Diagramme UML représentant l'édition d'une activité . . . . .	101
A.2	Diagramme UML représentant la planification d'une activité . . . . .	102

LISTE DES FIGURES

C.1	Grille présentant les reconnaissances . . . . .	107
C.2	Grille présentant l'émission des rappels . . . . .	108
C.3	Grille présentant le décompte des reconnaissances . . . . .	109

# Liste des tableaux

2.1	Répartition des différents capteurs dans l'appartement utilisés par PRADHA	33
3.1	Évaluation de la première méthode d'implantation . . . . .	40
4.1	Résultats de validation de l'algorithme de simple association . . . . .	49
4.2	Nombre de classes en fonction des activités . . . . .	58
4.3	Résultats de validation de l'algorithme de mixture de Gaussiennes . .	60
4.4	Avantages et inconvénients des deux algorithmes . . . . .	63
6.1	Programme détaillé de l'expérimentation . . . . .	78
7.1	Nombre de lancements . . . . .	82
7.2	Validation de l'algorithme . . . . .	82
7.3	Résultats des reconnaissances d'activité . . . . .	83
7.4	Niveaux d'importance des activités . . . . .	91

# Introduction

Au cours des dernières décennies, le monde occidental a connu de nombreux changements, dont deux qui pourraient avoir une forte influence sur notre quotidien : le vieillissement de la population et le développement de nouvelles technologies.

Le vieillissement de la population est un véritable problème car les personnes âgées sont les personnes les plus susceptibles d'avoir besoin d'aide notamment du fait de leur perte d'autonomie, les obligeant parfois à aller en maison de soins ou à avoir un(e) aide-soignant(e) à domicile.

D'un autre côté, les appareils électroniques sont et seront de plus en plus intégrés à notre quotidien, et ne cesseront d'évoluer dans les années à venir : grâce à la téléphonie mobile, nous sommes déjà connectés partout avec tout le monde, et les technologies se développent afin d'offrir de véritables outils d'assistance pour la maison, tels que des systèmes de réglage automatique de la température, d'ouverture des portes et des fenêtres, des lumières. Des écrans et autres systèmes d'interaction avec la maison seront bientôt disponibles dans de nombreux habitats. Grâce à ces équipements, l'informatique ubiquitaire, qui consiste en l'omniprésence d'appareils électroniques dans notre quotidien, deviendra une réalité dans les domiciles.

Pourquoi alors ne pas utiliser l'informatique ubiquitaire pour aider les personnes âgées, ou les personnes en manque d'autonomie en général ? Pourquoi ne pas rapprocher d'une part le problème de la perte d'autonomie et d'autre part la prolifération des appareils électroniques dans notre quotidien pour développer des outils d'aide à la réalisation de tâches de la vie quotidienne ? La perte d'autonomie se traduit principalement par des difficultés à réaliser les activités de la vie quotidienne (AVQ). Les AVQ sont des activités que l'on réalise tous les jours, comme se réveiller, faire à manger, manger, faire le ménage, aller aux toilettes *etc.*

## INTRODUCTION

Ces activités sont multiples et sont réalisées de façon automatique par les personnes saines. Cependant, les personnes en manque d'autonomie, en particulier les personnes atteintes de démence comme la maladie d'Alzheimer, ne réalisent plus ces activités aussi naturellement qu'avant. Elles peuvent avoir tendance à oublier de les faire, ou les faire partiellement, ou au contraire les faire plusieurs fois. Il faut alors trouver un moyen de leur rappeler efficacement les activités qu'elles ont à réaliser, sans toutefois les perturber en leur rappelant une activité déjà faite.

C'est à ce niveau qu'intervient la notion d'informatique ubiquitaire : en récupérant les informations provenant de l'appartement, il est possible de savoir ce que fait la personne. Une fois la connaissance de ce que fait la personne acquise, il devient possible d'adapter les rappels des AVQ en fonction de ce qu'elle fait, donc rappeler plusieurs fois une activité qui aurait été manquée et à l'inverse ne pas en rappeler une qui aurait été réalisée en avance ou à l'heure.

Il s'agit de savoir quelles activités rappeler, et à quel moment, pour pouvoir dire si la personne l'a bien faite ou non. Le moyen le plus simple et le plus instinctif pour savoir cela est d'établir un agenda dans lequel la personne entrerait ses AVQ avec leurs heures précises de réalisation. À partir de là, la maison intelligente est en mesure de savoir quelle activité rappeler à quel moment. Cela suppose bien sûr que l'agenda soit électronique, intégré à l'habitat, qui devra être un habitat intelligent, doté de divers capteurs, donc en accord avec la notion d'informatique ubiquitaire.

Ainsi, en mettant en place dans un habitat intelligent un calendrier électronique capable de rappeler les activités à réaliser dans la journée, il devient possible d'aider la personne dans la réalisation de ses AVQ.

Le laboratoire DOMUS<sup>1</sup> de l'Université de Sherbrooke est un laboratoire dédié à la recherche en informatique créant des technologies d'aide et de soutien aux personnes âgées souffrant de démence. Les problématiques étudiées dans ce laboratoire sont en lien avec ce qui a été mentionné plus haut, notamment l'assistance cognitive [35]. Les thèmes abordés par ce laboratoire sont à la fois des thèmes informatiques tels que la localisation dans l'appartement à partir de capteurs infrarouges, l'assistance mobile par un agenda intégré à un téléphone, mais aussi des thèmes en rapport à

---

1. [www.domus.usherbrooke.ca](http://www.domus.usherbrooke.ca)

## INTRODUCTION

la médecine tels que les soins à apporter aux personnes, le contact difficile par les personnes âgées avec des appareils électroniques, ou encore des thèmes en rapport avec le design d'interfaces. Cette pluridisciplinarité engendre de nombreux contacts des chercheurs du laboratoire avec des hôpitaux, des aides soignants, des ergothérapeutes, des designers *etc.*

Le projet de cette maîtrise s'inscrit dans le cadre de l'assistance cognitive en offrant un agenda aidant à la réalisation des AVQ. Il faut pour cela se pencher sur certains des thèmes abordés dans le laboratoire : la perte d'autonomie due aux troubles cognitifs, la conception d'habitats intelligents, l'algorithmique appliquée à la reconnaissance d'activité, la programmation orientée objet et le design d'interfaces.

Les personnes âgées ont la plupart du temps une certaine réticence à l'idée d'utiliser un ordinateur, un téléphone mobile ou tout autre appareil électronique. Elles n'ont pas l'habitude d'utiliser de tels appareils, d'où de nombreuses difficultés dans l'apprentissage de l'utilisation de telles technologies. Cet aspect est donc à prendre en compte pour concevoir un calendrier électronique qui soit utilisable par n'importe qui : il doit être à la fois instinctif, donc nécessitant peu d'apprentissage, et complet afin que la personne puisse enregistrer correctement ses activités.

Une fois les AVQ enregistrées sur le calendrier, la technologie mise en place dans l'appartement doit permettre de savoir ce qui s'y passe et en fonction de cela rappeler ou non l'activité à réaliser. Les informations provenant de capteurs répartis dans l'appartement sont récupérées et analysées et permettent alors de connaître l'activité en cours de réalisation. Le calendrier peut savoir s'il faut rappeler l'activité, et la rappeler le cas échéant.

Deux projets sont nécessaires pour mener à bien cette maîtrise : le projet consistant à implanter le calendrier électronique et celui consistant à implanter le programme servant à reconnaître les activités réalisées dans l'appartement. Le premier projet est appelé AMELIS (Alzheimer Mémoire et LIens Sociaux) et le deuxième projet est appelé PRADHA (Programme de Reconnaissance d'Activité Dans l'HAbitat). Ces deux projets doivent fonctionner en collaboration pour pouvoir atteindre les objectifs de cette maîtrise.

## INTRODUCTION

Le premier chapitre de ce mémoire expose l'état de l'art, au niveau des habitats intelligents et de la reconnaissance d'activité. Le deuxième chapitre reprend les objectifs et la méthodologie suivie pour mener à bien cette maîtrise. Le troisième chapitre décrit l'implantation et le fonctionnement du calendrier électronique, AMELIS. Il est suivi du chapitre quatre expliquant le fonctionnement de PRADHA. Le cinquième chapitre explique comment combiner les deux programmes pour qu'ils fonctionnent dans l'habitat intelligent. Une fois qu'il est possible de faire de la reconnaissance d'activité dans l'appartement, on souhaite pouvoir vérifier qu'elle fonctionne correctement. Une expérimentation a alors été menée et son protocole est résumé dans le sixième chapitre. Enfin, le septième et dernier chapitre décrit et interprète les résultats obtenus lors des expérimentations.



# Chapitre 1

## État de l'art

Ce chapitre a pour but de présenter les raisons qui nous ont poussé à faire ce projet de recherche. En premier lieu, il justifie à partir du contexte actuel, l'utilité d'un tel projet ainsi que ses applications. En deuxième lieu, les projets connexes liés à notre sujet de recherche sont présentés et permettent d'établir le cadre dans lequel s'inscrit notre projet. Enfin, en troisième lieu, sont présentées de façon détaillée les techniques généralement utilisées pour faire de la reconnaissance d'activité dans un appartement.

### 1.1 Un besoin pour les personnes âgées

#### 1.1.1 Une population vieillissante

Un des problèmes actuels de nos sociétés occidentales est le vieillissement de la population. Au Canada, environ 15% de la population est âgée de plus de 65 ans. De plus, il est prévu que d'ici 2020, une personne active sur cinq aura plus de 55 ans ([28]). Ceci soulève deux problèmes : d'une part, étant donné que la population vit de plus en plus vieille, de plus en plus de personnes seront en perte d'autonomie et ne pourront plus se débrouiller seules chez elles. Le deuxième problème qui survient est lié à la proportion personnes actives / inactives. Le nombre d'aide-soignants sera probablement insuffisant dans les années à venir pour pouvoir assumer tous les soins à apporter aux nombreux malades.

## 1.1. UN BESOIN POUR LES PERSONNES ÂGÉES

Globalement, les aînés ont le choix entre trois modes de vie : ou bien ils vivent chez eux (seuls ou avec leur conjoint/conjointe), ou bien ils vivent chez leurs enfants ou petits-enfants, ou encore ils vivent en maison de soins. En 2001, au Canada, 72,1% des aînés de plus de 65 ans vivent chez eux seuls ou avec leur conjoint/conjointe ([28]). Il s'agit donc d'une grande proportion d'aînés qui préfèrent rester chez eux plutôt que de partir en institution.

### 1.1.2 Le maintien à domicile

Le maintien à domicile des personnes âgées offre de nombreux avantages : les personnes sont chez elles, elles connaissent donc les lieux et sont habituées à leur environnement. Le fait qu'elles vivent seules ou à deux les pousse à rester autonomes et actives, contrairement aux maisons de soins où elles sont dépendantes des repas et des activités proposées par l'établissement. Elles peuvent garder les habitudes qu'elles ont toujours eues et sont encore capables d'accueillir des gens chez elles. Seulement, dans le cas où elles vivent seules, il y a danger pour leur santé si elles chutent ou oublient de réaliser leurs AVQ. En établissement, elles sont suivies et peuvent appeler de l'aide à tout moment, aide qui viendra très vite et aura les compétences pour leur porter secours.

Au Canada, parmi les 72,1% des personnes de plus de 65 ans vivant chez elles, 26,1% vivent seules, en dehors d'établissements ([28]). C'est ce type de population qui nous intéresse : des personnes suffisamment autonomes pour pouvoir vivre seules, mais pour lesquelles, étant donné leur âge élevé, il est risqué de vivre sans l'aide d'une autre personne. En effet, en 2003, un aîné âgé de 75 ans et plus sur dix et vivant dans un ménage privé avait besoin de l'aide d'une autre personne pour les soins de base tels que se laver, s'habiller, manger ou prendre des médicaments.

Le maintien à domicile est donc une solution confortable et préférable pour les personnes âgées. Cette solution s'inscrit d'autant mieux dans le contexte actuel en évitant la surcharge des centres de soins, mais peut être risquée s'il n'y a aucun suivi, ni de la part des parents, ni de la part d'aide-soignants. Avoir un aide-soignant toute la journée chez soi est une solution sécuritaire, mais très coûteuse, que peu de monde peut s'offrir. En effet, environ deux tiers des personnes restent chez elles et insatisfaites

## 1.1. UN BESOIN POUR LES PERSONNES ÂGÉES

de leurs soins attribuent cette insatisfaction à leur incapacité de payer [24]. De plus, comme il est dit plus haut, étant donné le nombre croissant d'aînés par-rapport au nombre de personnes actives, il n'y aurait pas suffisamment d'aide-soignants pour subvenir aux besoins de toutes les personnes âgées. Une solution serait donc d'alléger la tâche de l'aide-soignant en faisant appel à lui uniquement en cas de nécessité. Pour connaître ces cas de nécessité, il devrait être possible d'utiliser les informations récupérées à partir de l'habitat de la personne pour en déduire d'autres sur l'activité de cette personne. Ceci nous permet d'introduire la notion d'informatique ubiquitaire.

### 1.1.3 Informatique ubiquitaire

Quand Mark Weiser a introduit le paradigme de l'informatique ubiquitaire en 1991 [51], il pensait à un environnement où des personnes allant de leur bureau à chez eux pourraient toujours accéder à leurs données virtuelles, peu importe l'appareil utilisé. Il propose un environnement équipé de nombreux capteurs et ordinateurs qui apporteraient de multiples fonctionnalités et d'interactivité aux objets de la vie courante.

Certains chercheurs affirment que nous vivons déjà dans ce monde ubiquitaire grâce aux terminaux mobiles qui offrent accès à de grandes quantités d'informations provenant du web [3]. Il est dorénavant possible de prendre une photographie et de la partager instantanément avec tous ses amis, être au courant de tous les restaurants du quartier, ce qui nous permet de choisir celui qui correspond le mieux à ce qu'on cherche. Les informations de localisation permettent à nos appareils de nous guider à notre destination sans perdre de temps à regarder un plan de la ville.

Cependant, l'informatique ubiquitaire a de la difficulté à s'implanter à l'intérieur même de nos maisons. En effet, si elle est mobile et permet de récupérer des informations venant de n'importe où dans le monde, les informations provenant de l'intérieur de la maison sont quasiment inexistantes. Même les terminaux mis dans la maison ne servent pas à récupérer des informations de la maison, mais de l'extérieur.

Certaines entreprises comme Crestron<sup>1</sup> proposent des services d'automatisation telles que la fermeture automatique des volets, ou de réglage de la température, mais

---

1. [www.crestron.com](http://www.crestron.com)

## 1.1. UN BESOIN POUR LES PERSONNES ÂGÉES

cela va rarement plus loin. Les environnements ubiquitaires restent confinés dans des ordinateurs, des tablettes ou des téléphones mobiles sans être avertis de ce qui se passe à l'intérieur de la maison.

Un environnement ubiquitaire doit être en mesure de faciliter notre vie, ou bien en nous assistant dans la réalisation de certaines tâches, ou bien en s'occupant de certaines tâches pour nous ou encore en nous les rappelant. Le but des environnements ubiquitaires est de permettre des interactions entre une personne et l'environnement intelligent via des moyens de communication entre les réseaux sociaux, les amis, la famille et les autres.

Les trois aspects essentiels pour un déploiement efficace de services contextuels dans un environnement intelligent peuvent être résumés par : la capacité d'un système à capter des événements, la compréhension de ces événements et les services offerts par le système. En développant l'informatique ubiquitaire dans les maisons pour en soustraire des informations sur ce qui s'y passe, on serait donc en mesure d'offrir une assistance à ses occupants. L'informatique ubiquitaire permettrait de répondre à de nombreux besoins, tous différents en fonction des personnes vivant dans l'habitat intelligent. Certains aimeront programmer dans leur calendrier des événements particuliers ou occasionnels, comme des rendez-vous, et souhaiteraient un ou plusieurs rappels. D'autres aimeraient que la machine à café démarre automatiquement à une heure précise pour que le café soit prêt au moment du déjeuner. D'autres encore utiliseraient leur téléphone pour régler de leur lieu de travail la température de la maison ou pour allumer automatiquement la lumière pour donner une impression d'occupation la nuit tombée.

Dans notre contexte, il s'agirait de rappeler des AVQ et de s'assurer avec un minimum d'assurance qu'elle a bien été réalisée. Pour la plupart des personnes, il y aurait peu d'intérêt à rappeler les AVQ, car elles se réalisent « naturellement » : on mange quand on a faim, on se lave au réveil avant d'aller travailler, on fait la vaisselle une fois le repas terminé *etc.* Cependant, pour les personnes âgées vivant seules, il se peut que certaines activités qui paraissent naturelles ne soient pas réalisées à temps, voire pas du tout. C'est donc vers ce genre de situations que se dirige notre problématique :

## 1.2. LES MOYENS DE RAPPEL EXISTANTS DANS UN ENVIRONNEMENT INTELLIGENT

le rappel efficace d'AVQ.

### 1.2 Les moyens de rappel existants dans un environnement intelligent

Plusieurs études [45] ont traité du même problème du rappel d'activité dans un habitat intelligent, dans le but d'offrir une assistance à la personne. Il existe deux types d'assistance : l'assistance cognitive et l'assistance physique. L'assistance physique consiste à aider la personne à surmonter un problème physique par différents moyens comme un monte-charge ou des commandes à distance. L'assistance cognitive, qui est l'assistance qui nous intéresse, consiste à aider la personne à pallier des déficits cognitifs dus à l'âge ou la maladie. Cette aide est sous la forme de sons émis dans l'appartement, ou bien de vidéos diffusées sur des écrans répartis dans l'appartement ou de messages envoyés sur un cellulaire ou tout autre moyen de communication. Le premier principe de cette assistance consiste en l'envoi d'un message à la personne, message qui doit servir à l'aider. Ce message peut être sous la forme d'un ordre, d'un rappel, d'un tutoriel, d'une question *etc.*

Dans cette section vont être présentés les différentes techniques existantes dans la littérature permettant de faire des systèmes de rappels et des systèmes d'assistance intelligents.

La plupart des gens utilisent des moyens de rappel dans leur vie quotidienne, moyens de rappels qui peuvent être sous différentes formes. Avant le développement des calendriers électroniques, ils étaient sous la forme d'agendas papier ou de calendriers papier muraux. Les gens avaient pour habitude de les consulter régulièrement, ce qui leur évitait de manquer leurs rendez-vous. Par-contre, jamais personne ne planifie dans son calendrier mural des AVQ, étant donné que le calendrier n'est pas en mesure d'émettre de rappels.

L'idée d'agendas électroniques n'est pas une idée récente. En 1997 paraissait un des premiers agendas électroniques [27] permettant de générer automatiquement un programme pour la journée en fonction des activités à faire. Ces activités comprennent

## 1.2. LES MOYENS DE RAPPEL EXISTANTS DANS UN ENVIRONNEMENT INTELLIGENT

des activités obligatoires telles que faire sa toilette ou déjeuner, mais aussi des activités facultatives qui arrivent de façon ponctuelle et peuvent obliger à réaménager l'agenda habituel afin de les insérer.

Aujourd'hui, avec les agendas électroniques, il est possible d'émettre des rappels à chaque fois qu'il est l'heure de faire une activité. Il est donc envisageable d'utiliser un agenda électronique pour se faire rappeler des AVQ, chose impossible sans la technologie. Cependant, avec les agendas « classiques » tels que GoogleAgenda<sup>®</sup><sup>2</sup> ou iCal pour Apple<sup>®</sup><sup>3</sup>, il s'agit à chaque fois de rappels émis à des heures fixes, ce qui limite grandement la flexibilité et l'utilité d'un tel agenda dans le rappel des AVQ car les gens ne vivent en général pas en fonction d'un planning fixe. En effet, concernant les AVQ, la plupart des utilisateurs vont préférer planifier des activités relativement à d'autres activités. Ils vont par-exemple planifier l'activité de faire la vaisselle immédiatement après manger et non à une heure fixe. Ainsi, le rappel de l'activité de faire la vaisselle devrait être émis en fonction de la durée du repas.

C'est ce que cherche à faire l'agenda Autominder [37, 39]. Il s'agit d'une orthèse cognitive fournissant à son utilisateur un moyen de rappel pour les activités quotidiennes en fonction de ce qui est fait et est à faire.

Il est décomposé en trois modules [36]. Un premier module qui connaît l'agenda de l'utilisateur, donc ce qu'il est sensé faire. Un deuxième module qui est utilisé pour récupérer les informations provenant de l'appartement et ainsi savoir ce qu'est en train de faire l'utilisateur. Enfin, le dernier module sert à prendre la décision de ce que qu'il faut rappeler à l'utilisateur pour s'assurer qu'il le fasse correctement. C'est ce dernier module le cœur du programme ; il utilise des techniques d'intelligence artificielle comme les réseaux bayésiens pour mettre à jour automatiquement l'agenda en fonction de ce qui a été réalisé et surtout en fonction du programme de la journée. Il doit balancer entre quatre buts pour savoir quoi rappeler : s'assurer que l'utilisateur est au courant de ce qu'il doit faire, lui donner satisfaction, éviter qu'il fasse mal les activités et enfin éviter qu'il accorde trop de confiance au système, au point de perdre toute autonomie dans sa prise de décision.

---

2. <https://www.google.com/calendar>

3. [www.apple.com](http://www.apple.com)

## 1.2. LES MOYENS DE RAPPEL EXISTANTS DANS UN ENVIRONNEMENT INTELLIGENT

Autominder est actuellement disponible sur un robot mobile (Pearl, [38]) capable à partir de caméras, de capteurs et de microphones, de connaître l'activité qu'est en train de faire l'utilisateur et ainsi adapter son agenda.

Autominder est similaire à l'agenda PEAT [27], qui fut le premier agenda commercialisé à offrir un système de planification flexible et adapté. Il s'agit d'un agenda électronique mobile qui fonctionne à partir de routines pré-enregistrées par l'utilisateur. Au cours de la journée, l'agenda indique à l'utilisateur l'activité à faire et lui demande s'il souhaite la réaliser ou s'il préfère la reporter ou l'abandonner. En fonction de la réponse, l'agenda devra se mettre à jour pour répondre au mieux aux demandes de l'utilisateur.

La différence principale entre PEAT et Autominder vient de leur moyen pour savoir l'activité qu'est en train de faire l'utilisateur : PEAT demande à chaque fois ce que l'utilisateur est en train de faire et ce qu'il souhaite faire ensuite, alors que Autominder va être en mesure de le savoir par lui-même. Autominder fonctionne donc de façon beaucoup plus automatique et autonome que PEAT ; il n'est cependant pas mobile, ce qui empêche l'utilisateur de l'emporter partout avec lui, à l'extérieur notamment.

COACH [29] est un outil destiné à aider les personnes âgées atteintes de démence dans la réalisation d'une tâche bien précise, nécessitant plusieurs étapes indispensables. Il utilise différentes techniques de vision et d'intelligence artificielle pour identifier l'action qu'est en train de faire l'utilisateur et en fonction de cela lui offrir une assistance vocale et/ou visuelle.

COACH est actuellement capable de superviser et d'assister l'utilisateur dans la tâche de lavage de mains. Il va être en mesure de vérifier que les étapes « ouvrir le robinet d'eau », « mettre du savon », « se rincer les mains », « se sécher les mains » et « couper l'eau » sont bien exécutées dans l'ordre. Il est également possible de modifier l'ordre de certaines étapes, comme de mettre du savon avant d'ouvrir le robinet. Il s'agit donc d'un système de guide à la réalisation d'activité qui, tout comme Autominder, va chercher à offrir une assistance à la personne en fonction de ce qu'elle est en train de faire.

## 1.2. LES MOYENS DE RAPPEL EXISTANTS DANS UN ENVIRONNEMENT INTELLIGENT

Le projet Archipel [1] cherche également à offrir une assistance à la personne pour la réalisation d'une tâche précise. La tâche choisie est une tâche domiciliaire complexe : réalisation d'une recette dans la cuisine. Archipel fournit une assistance dans la mesure où il indique à l'utilisateur les tâches ou plus exactement les actions à faire. Par exemple, pour aider l'utilisateur à réaliser un pâté chinois, Archipel va faire plusieurs choses :

- afficher les étapes de la recette sur un écran tactile, les unes après les autres
- indiquer pour chaque étape sur l'écran tactile où se trouvent les ustensiles
- si cela ne suffit pas, une lumière va s'allumer sur le placard pour indiquer où récupérer l'ustensile requis
- vérifier que le bon placard a été ouvert pour récupérer l'ustensile nécessaire à la réalisation de l'étape
- valider l'étape de l'activité si toutes les sous-étapes ont été faites correctement et ainsi passer à l'étape suivante.

Il s'agit donc clairement, tout comme COACH, d'un assistant à la réalisation d'une tâche précise, avec de la reconnaissance d'activité. L'utilisateur ne peut pas passer à l'étape suivante tant qu'il n'a pas validé l'étape en cours. L'activité est représentée sous la forme d'un arbre hiérarchique où chaque nœud infère si sa tâche est réussie. L'information provenant des capteurs de l'environnement est transmise des feuilles de l'arbre jusqu'au nœud responsable de l'étape en cours. Grâce aux inférences de chaque nœud, une étape peut être réussie quel que soit l'ordre des tâches, du moment que celles obligatoires sont réalisées.

Cependant, contrairement à COACH, l'utilisateur n'est pas filmé, ce qui rend Archipel moins intrusif que COACH et donc plus agréable à une utilisation quotidienne. De plus, tout comme COACH, Archipel laisse tout d'abord l'utilisateur faire par lui-même l'activité avant de lui apporter une quelconque assistance. Il s'agit d'un principe de base pour éviter que l'utilisateur devienne trop dépendant d'un tel outil et puisse garder une certaine forme d'autonomie.

Yi Chu présente dans [10] un système de reconnaissance d'activité et de rappel utilisé pour faire de l'assistance à la personne. Il a pour but d'aider la personne à



## 1.2. LES MOYENS DE RAPPEL EXISTANTS DANS UN ENVIRONNEMENT INTELLIGENT

suivre un programme précis qui aura été au préalable enregistré et peut ainsi, à partir de données provenant de capteurs, savoir s'il doit émettre un rappel ou non. L'outil PUCK [13, 12] est un outil similaire qui identifie quand une étape d'une activité a été manquée ou mal faite et va donc émettre un rappel si nécessaire. Ils utilisent tous deux pour savoir si l'activité a été manquée ou erronée des techniques de forage de données et d'apprentissage automatique à partir de données capteur et d'étiquettes RFID. Tous les deux ont pour but d'aider la personne à exécuter correctement les activités qu'elle a prévu de faire.

Yi Chu défend dans [10] qu'il faut émettre le moins de rappels possible pour éviter de perturber l'utilisateur, donc par-exemple dans des situations où l'activité prévue n'a pas encore été réalisée. Les rappels doivent de plus être émis au moment opportun, c'est-à-dire lorsque l'utilisateur n'est pas en train d'effectuer une activité plus importante que l'activité à rappeler. À l'inverse, PUCK émet de nombreux rappels en fonction de ce que fait la personne. Les rappels sont émis dans les situations suivantes : si l'utilisateur fait une activité autre que celle demandée, si une des étapes de l'activité à faire est mal faite et si l'utilisateur met trop de temps à faire une étape. PUCK défend donc l'utilisation de rappels multiples pour forcer l'utilisateur à réaliser ses activités, alors que le programme de Yi Chu va s'adapter à ce que fait l'utilisateur pour rappeler efficacement les activités à réaliser.

Ainsi, on se retrouve avec de nombreux agendas électroniques et systèmes d'assistance, chacun répondant à des problèmes différents. COACH et Archipel sont des outils d'assistance à la réalisation d'activité, mais ne sont en aucun cas des agendas électroniques. Autominder et PEAT sont des agendas électroniques, mais ne permettent pas d'assister la personne dans la réalisation de ses activités, qu'elles soient planifiées ou non. PUCK permet d'assister les personnes dans la réalisation de leurs AVQ, mais n'utilise pas les informations provenant d'un agenda électronique pour savoir quelles activités assister.

Finalement, le programme proposé par Yi Chu est le système qui correspond le plus à ce que l'on souhaite faire puisqu'il s'agit d'un calendrier qui s'adapte à ce que fait l'utilisateur et émet un rappel seulement s'il estime qu'il est nécessaire. Cependant, le système en question diffère en deux points de ce que l'on souhaite. D'une

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

part, il utilise, en plus des capteurs de l'appartement, des étiquettes RFID permettant d'identifier les objets que portent les utilisateurs. Ceci peut être contraignant pour l'utilisateur et peu fiable si le marqueur en question n'est pas mis en place correctement. D'autre part, un des buts du programme de Yi Chu est d'offrir une assistance à l'utilisateur qui lui soit non-contraignante, c'est-à-dire qui s'adapte à ce qu'il fait, quitte à manquer une activité. Or, notre but est de s'assurer que la personne réalise bien l'activité prévue, et non qu'elle la réalise seulement si elle le désire.

## 1.3 Reconnaissance d'activité dans un habitat

De nombreux laboratoires de par le monde se sont penchés sur le problème de la reconnaissance d'activité, que ce soit à partir d'informations provenant de caméras ou de capteurs divers et variés. Nous tenterons dans ce chapitre de traiter des différentes méthodes qui ont déjà été employées dans la littérature pour faire de la reconnaissance d'activité dans des habitats intelligents, et introduirons celle que nous avons choisie.

### 1.3.1 Reconnaissance à partir de caméras

Comme il a été vu avec les cas de COACH et de Autominder, il est possible de faire de la reconnaissance d'activité à partir de données provenant de caméras. Depuis les dernières années, la reconnaissance d'activité par caméra a été un pan de la recherche très important. En effet, on retrouve dans notre vie courante de plus en plus de caméras, que ce soit dans la rue, les immeubles, chez soi... Ceci engendre des quantités énormes de vidéos à traiter. L'intérêt de la reconnaissance de mouvements tout d'abord, d'activité ensuite, prend alors tout son sens : en automatisant le processus d'analyse des vidéos, on peut en tirer des informations pertinentes.

Dans notre vie quotidienne seront intégrés sans qu'on s'en rende compte de nombreux appareils électroniques avec lesquels il sera possible d'interagir [34]. Maja Pantic dans [34] insiste sur le fait que pour interagir correctement avec un ordinateur, chose que tout le monde aura à faire continuellement le long de la journée, des caméras seront disposées partout dans l'appartement. Il faut que les caméras puissent répondre aux questions suivantes : qui est l'utilisateur ? où est-il ? que fait-il ? comment le

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

fait-il ? quand ça ? pourquoi ? Ceci rejoint les buts de l'informatique ubiquitaire avec l'omniprésence des ordinateurs dans notre vie quotidienne. Toutefois, lorsque l'on souhaite implanter de tels systèmes dans des maisons, la plupart des algorithmes de reconnaissance d'activité évitent l'utilisation de caméras et de microphones, pour des raisons de respect de la vie privée.

Pour ce qui concerne la reconnaissance d'une activité complexe, il va falloir décomposer cette activité en plusieurs étapes que l'on va analyser successivement. Une fois une étape repérée, on applique différentes techniques de reconnaissances de formes pour bien l'identifier. L'analyse de trajectoires et de mouvements peut répondre à ce problème, en utilisant des chaînes de Markov [32]. Les chaînes de Markov, qu'elles soient cachées ou non, dynamiques ou non, sont des techniques d'apprentissage couramment utilisées pour faire de la reconnaissance d'activité à partir de vidéos. À partir d'une fenêtre d'analyse, elles vont permettre de modéliser le passage d'une action à une autre, passage nécessaire à la réalisation de l'activité.

#### 1.3.2 Reconnaissance à partir de données des capteurs

Comme on a pu le voir en section 1.3.1, la reconnaissance d'activité à partir de caméras peut s'avérer utile pour reconnaître des activités de courte durée, ou encore des mouvements précis tels que des chutes. Cependant, le problème du respect de la vie privée, dû à l'enregistrement en continu de ce que font les personnes, enregistrement qui pourrait être éventuellement visionné par d'autres personnes, empêche d'utiliser les caméras comme moyen de reconnaissance d'activité.

Cette section va donc se consacrer à établir une revue de littérature sur les techniques utilisées pour faire de la reconnaissance d'activité à partir de données de capteurs. Cette reconnaissance peut être divisée en deux techniques majeures ([43, 7]) : approche par analyse de données (*data-driven approach* en anglais) comme des réseaux bayésiens, des chaînes de Markov *etc.* et approche par analyse de connaissances (*knowledge-driven approach* en anglais) comme des techniques de modélisation et raisonnement logique, telles que les ontologies.

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

#### **Approche par analyse de connaissances**

Dans le cas de l'approche par analyse de connaissances, le modèle d'activité se base sur la façon dont une activité doit être faite. Il est généré à partir de la modélisation de la logique des raisonnements à avoir pour réaliser l'activité. Le principe est de créer une bibliothèque de toutes les activités réalisables dans l'appartement, d'analyser les informations fournies par les capteurs puis en déduire le raisonnement logique à avoir pour faire cette activité. Cette méthode permet de faire de la reconnaissance de plan [4] en identifiant les événements des capteurs nécessaires pour créer ce qu'on appelle « le plan d'une activité » et ainsi être en mesure de prédire ce que souhaite faire l'utilisateur. Les raisonnements logiques permettent d'avoir une sémantique claire et lisible par n'importe qui ; il est également possible de facilement intégrer des informations supplémentaires comme des informations concernant l'environnement et l'utilisateur et ainsi fusionner ces données pour obtenir des informations plus précises et fiables. Un défaut de la représentation logique est son manque d'adaptabilité et son incapacité à gérer l'incertitude [6].

Une des méthodes utilisée pour faire des raisonnements logiques est ce qu'on appelle les ontologies. Il s'agit d'une méthode consistant à décrire les objets, les événements, les ressources, les contraintes afin de produire des consignes assignant les ressources et les temps aux objets et aux événements [20]. Par-exemple, pour modéliser l'activité de préparer à manger, on associerait des objets tels que des couteaux, des fourchettes, des poêles, des assiettes ; des événements tels qu'allumer le four, ouvrir le réfrigérateur ; des ressources comme des aliments ; des contraintes telles qu'ouvrir le robinet d'eau, allumer la lumière *etc.* Tous ces éléments permettent d'établir des règles et ainsi modéliser l'activité de préparer le déjeuner. Il est facile d'ajouter ou de supprimer des données sur le modèle, offrant dans ce cas au modèle une certaine flexibilité.

L'utilisation d'ontologies permet donc de spécifier la sémantique d'une activité, et le raisonnement bâti autour de toutes les activités permet de les identifier dans l'environnement. L'utilisation d'ontologies offre de nombreux avantages dans la reconnaissance d'activité dans un appartement intelligent ; cependant, une étude présentée dans l'article [43] montre que leur efficacité est encore à prouver comparé à l'utilisation de l'approche par analyse de données. En effet, à moins d'y intégrer des

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

informations temporelles et d'utiliser des techniques d'apprentissage automatique revenant finalement à faire de l'analyse de données, les résultats sont peu convaincants.

Étant donné que l'on souhaite un programme de reconnaissance d'activité qui soit efficace et peu compliqué à mettre en œuvre, c'est vers des techniques « plus traditionnelles », comme l'analyse de données, que nous allons nous tourner.

#### **Approche par analyse de données**

Dans le cas de l'approche par analyse de données, de nombreuses études ont été faites, toujours poursuivant le même but : obtenir le meilleur taux de reconnaissance, pour des activités toujours plus complexes, toujours plus difficiles à identifier simplement à l'aide de capteurs binaires dispersés dans un appartement.

Le terme *données* se différencie du terme *connaissances* dans la mesure où *données* correspond aux données brutes des capteurs, alors que *connaissances* signifie la connaissance de ce qu'est une activité, ce qui peut être la démarche à suivre pour la réaliser, ou sa définition par rapport aux autres activités réalisables dans l'appartement. Maintenant que nous avons vu rapidement cette procédure, nous allons voir comment utiliser les données fournies par les capteurs peuvent nous permettre de modéliser des activités.

Tout d'abord, deux techniques majeures d'apprentissage permettent de générer les modèles d'activité : apprentissage supervisé et apprentissage non supervisé.

**Apprentissage non supervisé :** Si on choisit de faire un apprentissage non supervisé, on suppose alors que des schémas extraits automatiquement à partir d'évènements capteur vont décrire correctement les activités sans aucune connaissance de ce qu'est l'activité [9]. Cela signifie qu'en entrant dans le programme d'apprentissage de simples données de capteurs, générées dans l'appartement pendant une durée  $t$ , le programme va être capable de regrouper certains évènements qui apparaissent fréquemment en même temps et leur assigner une classe. Par-exemple, si on analyse les activités se déroulant pendant une semaine dans un appartement, on remarque plusieurs fois les activités suivantes : préparer le thé, manger, se laver, sous forme d'évènements de capteurs. L'algorithme va être capable, à partir de toutes ces données, d'établir trois classes pour lesquelles certains évènements apparaissent de façon

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

récurrente. Ceci est possible parce que les activités ne sont pas répétées à la même fréquence, ni dans le même ordre : on aura tendance à se doucher une seule fois par jour mais à s'habiller ou à préparer du thé plusieurs fois dans une même journée, ce qui permet au bout de la semaine de les différencier. Ainsi, à la fin de la phase d'apprentissage, on obtient trois modèles d'activité. Si on exécute l'algorithme pour faire de la reconnaissance et que dans l'appartement la suite d'évènements générés ressemble à celle générée lors de la réalisation de l'activité de préparer le déjeuner, alors c'est cette activité qui sera reconnue.

L'avantage principal de cette technique [11, 52] est d'éviter le processus long et laborieux d'annotation des données de capteurs afin d'obtenir un jeu de données où chaque séquence d'évènements correspond à une activité précise. Cependant, comme il a été mentionné dans [11], un apprentissage non supervisé à partir de données de capteur présente un certain nombre de défis, en particulier lorsque l'on dispose de peu d'instances d'activités. Le bruit généré lorsque la personne réalise l'activité est difficile à discerner de la donnée utile. De plus, le fait qu'une activité puisse être faite de différentes façons peut engendrer la création de deux modèles différents pour la même activité, alors que nous souhaitons un seul modèle. Enfin, étant donné que nous disposons au sein du laboratoire d'un jeu de données déjà annoté qui a été généré lors de précédentes expérimentations, la solution utilisant un apprentissage supervisé devient la plus rapide à implanter et la plus efficace dans notre cas.

**Apprentissage supervisé :** Un apprentissage supervisé requiert que tous les évènements capteurs soient annotés pour déterminer l'activité qui les a générés.

En apprentissage supervisé, on dispose de données annotées que l'on utilise pour l'apprentissage de modèles d'activités. La procédure usuelle pour établir les modèles de reconnaissance d'activité se divise en trois phases :

1. Phase de collecte des données des capteurs :
  - (a) recueillir les données de capteurs générés pendant qu'une personne réalise une activité précise
  - (b) annoter les évènements des capteurs pour indiquer ce que fait la personne et à quelle heure

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

- (c) répéter les étapes précédentes plusieurs fois pour entraîner l'algorithme (c'est-à-dire demander plusieurs fois à la personne de réaliser la même activité, que l'on annote.) Ce nombre dépend de l'algorithme choisi et est à déterminer en fonction des résultats de l'apprentissage : un nombre trop important d'instances entraînerait un sur-apprentissage, un nombre trop faible entraînerait un sous-apprentissage. Une quinzaine d'instances par activité serait un minimum.
2. Phase d'apprentissage :
    - (a) déduire à partir des données recueillies une représentation pour chaque activité
    - (b) répéter l'étape précédente plusieurs fois pour entraîner l'algorithme et ainsi obtenir une représentation fiable. Le nombre de fois nécessaire pour entraîner correctement l'algorithme dépend de la convergence des paramètres des modèles et peut varier entre cinq et deux-cent itérations.
  3. Phase de validation : vérifier que l'algorithme fonctionne correctement sur de nouvelles données, donc des données autres que celles de l'apprentissage.

Il existe de nombreux algorithmes pour modéliser une activité grâce à un apprentissage supervisé : les modèles de Markov cachés (MMC), les réseaux bayésiens dynamiques et naïfs, les arbres de décision, l'algorithme des  $k$  plus proches voisins et les machines à vecteur de support. Les MMC et les réseaux bayésiens sont les techniques les plus utilisées pour la reconnaissance d'activité [43].

Les MMC utilisent des techniques d'apprentissage qui, contrairement aux réseaux Bayésiens, incluent des informations temporelles [5]. Un MMC est un modèle probabiliste composé de variables cachées et de variables observables à chaque temps  $t$ . Les variables cachées correspondent à l'activité  $A$  que l'on souhaite reconnaître, les variables observables correspondent aux données provenant des capteurs  $x$  [48]. Ainsi, d'après la théorie de Markov, on suppose que l'activité  $A_t$  dépend uniquement de son état précédent  $A_{t-1}$ . De même, il faut que  $x$  (les données des capteurs à l'instant  $t$ ) dépende uniquement de  $A_t$ . Il devient alors possible de calculer la probabilité de passer d'un état à un autre, donc d'une activité à une autre, étant donné les événements générés dans l'appartement.

### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

Le fonctionnement de l'algorithme est donc assez simple et facile à mettre en œuvre. De plus, les MMC requièrent en général peu de jeux de données pour la phase d'apprentissage. Cependant, comme mentionné dans [49], les MMC ne sont pas adaptés pour reconnaître des activités longues à réaliser, celles qui laissent la place à plusieurs réalisations différentes et celles dont les performances ont tendance à se dégrader [15].

Le Classificateur Bayésien Naïf (CNB) est une technique très répandue pour la reconnaissance d'activité [47]. Il obtient parfois de bons résultats de reconnaissance malgré le fait qu'il suppose l'indépendance conditionnelle entre les variables des attributs et n'inclut aucune information temporelle. Le classificateur calcule la classe la plus probable étant données les informations reçues en entrée en utilisant la règle de Bayes [47].

Étant donné la simplicité du modèle obtenu par le classificateur Bayésien naïf, les résultats sont peu satisfaisants et donc nécessitent l'ajout d'informations temporelles [47]. Pour ce faire, il faut ajouter lors de l'apprentissage deux informations :

- Information 1 : l'activation d'un certain capteur a eu lieu pendant la réalisation de l'activité.
- Information 2 : le capteur  $C_1$  a été activé avant le capteur  $C_2$ . Ceci traduit le fait qu'un certain ordre est attendu dans la réalisation des tâches d'une activité.

Les résultats obtenus permettent de reconnaître des AVQ telles que la prise de médicaments, la préparation et la prise d'un repas, le travail devant un ordinateur *etc.*

Un problème rencontré avec les CNB apparaît dans le cas où de nombreux capteurs ne sont jamais activés, comme cela peut arriver pour les activités courtes ou nécessitant peu d'actions de la part de l'utilisateur. En effet, si une activité est principalement décrite par des capteurs non-activés, la probabilité qu'elle se réalise sera toujours nulle et elle ne sera donc jamais reconnue. Cependant, il est possible d'appliquer des techniques de lissage pour améliorer ces résultats [44]. Ainsi, lorsqu'une activité n'active que peu de capteurs, grâce au lissage elle n'obtiendra pas une probabilité nulle concernant sa réalisation. Le lissage va permettre d'utiliser les données provenant de tous les capteurs de l'appartement, même ceux qui n'ont pas été activés, pour leur affecter un certain poids ; ce poids est déterminé par-rapport aux capteurs



### 1.3. RECONNAISSANCE D'ACTIVITÉ DANS UN HABITAT

activés de façon générale dans l'appartement. Cette technique améliore considérablement les résultats pour des activités courtes et simples.

Dans notre contexte, nous souhaitons savoir si une activité  $A$  a été réalisée pendant un court intervalle de temps, intervalle déterminé par AMELIS. Les différentes façons de réaliser l'activité  $A$  doivent être prises en compte pour construire le modèle d'activité mais ne doivent pas permettre d'identifier comment elle a été réalisée. En effet, nous ne souhaitons pas connaître le détail de réalisation d'une activité, seulement savoir si elle a été réalisée, et donc la valider si elle a été faite correctement, peu importe comment elle a été faite. Ainsi, un programme de reconnaissance d'activité utilisant des méthodes de partitionnement de données, *clustering* en anglais, devrait pouvoir répondre à notre problème. Il s'agit d'une méthode qui va tâcher pour chaque activité de rassembler dans différentes classes les événements des capteurs qui sont générés souvent ensemble. Ceci résulte en la création de plusieurs classes d'activité au sein même d'une activité. Ce partitionnement de données devrait être en mesure d'inclure les différentes façons d'exécuter correctement l'activité en question.

Tâm Huỳnh dans [25] distingue deux types d'activités :

- les activités de bas niveau, c'est-à-dire des activités courtes, typiquement durant entre une dizaine de secondes et plusieurs minutes. Ce serait par-exemple marcher, se lever, manger une collation *etc.*
- les activités de haut niveau, qui regroupent plusieurs activités de bas niveau et peuvent durer de plusieurs minutes à plusieurs heures. Préparer un repas, faire le ménage, faire la vaisselle seraient des activités de haut niveau.

Ainsi, en utilisant ces notions, on est en mesure de dire que COACH et PUCK vont chercher à faire de la reconnaissance de bas niveau, comme le brossage de dents, alors que des programmes tels que Autominder et celui de Yi Chu vont chercher à reconnaître des activités de haut niveau, telles que préparer un repas. La méthode de partitionnement de données est justifiée principalement pour les activités de haut niveau.

## 1.4. BILAN

### Détermination de la fenêtre de reconnaissance

Faire de la reconnaissance d'activité dans un appartement implique de tout d'abord connaître la fenêtre de reconnaissance à analyser. Habituellement, cette fenêtre inclut deux informations : le début de l'activité et la durée de l'activité, ce qui permet de définir la largeur de la fenêtre et son commencement [26].

[26] propose une fenêtre dynamique glissante de reconnaissance qui s'ajuste en fonction de l'activité à reconnaître. Au lieu de définir une fenêtre fixe statique, sont définis les événements qui vont servir à définir les limites de la fenêtre dynamique. Il faut alors connaître les événements qui vont permettre de définir le début et la fin de la fenêtre. Ces événements peuvent être choisis comme étant les événements *révélateurs* d'une activité, c'est-à-dire les événements qui sont indispensables au commencement et à la fin d'une activité. Cette méthode permet d'avoir une fenêtre d'une taille dynamique. Cette fenêtre est également glissante car elle se déplace au cours du temps. Ainsi, une nouvelle fenêtre sera générée à chaque fois qu'un nouvel événement sera déclenché dans l'appartement. Cette méthode permet d'obtenir de bons résultats de reconnaissance, proches de 70%.

Dans [21], les fenêtres glissantes sont utilisées pour reconnaître des activités simples, concurrentes ou entrelacées. Ces fenêtres glissantes sont combinées avec un système de pondération qui forme le cœur du programme présenté dans l'article.

## 1.4 Bilan

De nos jours, deux motivations principales nous poussent à créer un calendrier interactif capable de reconnaître les activités réalisées dans l'appartement :

- il y aura de plus en plus besoin de systèmes d'assistance à la personne électronique dû à l'augmentation du nombre de personnes âgées et au manque de personnel soignant ;
- les appartements commencent tout juste à être *intelligents*, c'est-à-dire à incorporer des systèmes d'écrans et de capteurs pour s'adapter aux besoins des usagers, permettant de soulager les tâches domiciliaires, et donc assister les personnes dans leur vie quotidienne.

#### 1.4. BILAN

De nombreux systèmes d'assistance existent déjà, que ce soit sur le marché ou en tant que protocoles expérimentaux : agendas qui s'adaptent en fonction de ce que fait la personne, systèmes d'assistance à la réalisation d'une tâche précise, systèmes de rappel intelligent *etc.* Cependant, aucun ne permet, de façon non-intrusive, d'aider la personne dans la planification de sa journée en ne lui rappelant les activités à faire que si elles n'ont pas été réalisées.

Pour un tel système, un programme de reconnaissance d'activité de haut niveau est nécessaire. De nombreux programmes de reconnaissance d'activité ont déjà été implantés, que ce soit pour la surveillance vidéo, le suivi d'activité ou l'assistance. La plupart sont capables de reconnaître des activités de bas niveau, en portant attention à l'ordre des tâches pour mener à bien cette activité, à l'aide notamment des chaînes de Markov. Certaines utilisent les classificateurs Bayésiens naïfs mais obtiennent des résultats peu probants s'ils ne sont pas combinés avec des lisseurs ou des informations temporelles. Enfin, en particulier pour les activités de haut niveau, des techniques de partitionnement de données donnent de bons résultats.

# Chapitre 2

## Objectifs et méthodologie

L’objectif général de la maîtrise est de créer un calendrier électronique capable de vérifier si les activités planifiées ont bien été réalisées dans l’appartement, et de les rappeler si nécessaire. Cela signifie qu’il faut d’une part créer un calendrier électronique, AMELIS, et d’autre part créer un programme de reconnaissance d’activité qui soit capable de communiquer avec l’appartement, PRADHA.

Ce chapitre détaille ces deux objectifs ainsi que la méthodologie suivie pour les réaliser. Ces objectifs se situent d’abord au niveau de la programmation, aussi bien pour AMELIS que pour PRADHA, mais également au niveau de l’utilisabilité avec AMELIS, et de l’algorithmique, avec PRADHA.

### 2.1 Objectifs liés à AMELIS

#### 2.1.1 Objectifs au niveau des fonctionnalités

AMELIS a été conçu afin d’être utilisable par des personnes âgées, atteintes de démence. Il faut donc que l’interface soit conviviale, simple d’utilisation et minimise le nombre d’étapes nécessaires à l’utilisateur pour entrer ses activités. En effet, on souhaite que l’utilisateur, peu importe ses compétences en informatique, soit capable par lui-même de planifier ses activités. De plus, le calendrier a pour objectif d’offrir d’autres fonctionnalités telles que la gestion de contacts, d’adresses, de médications et d’images. Ces fonctionnalités ont pour but d’aider l’utilisateur à gérer son quotidien

## 2.2. OBJECTIFS LIÉS À PRADHA

et à se remémorer les évènements passés.

### 2.1.2 Objectifs au niveau de la programmation

Un autre objectif poursuivi par AMELIS porte sur la programmation elle-même. Le langage choisi est Java. Étant donné qu'AMELIS va être utilisé par d'autres programmes, tel que PRADHA, il est nécessaire qu'il soit codé de façon modulaire, donc qu'il soit composé de plusieurs modules utilisables par différents programmes. Ces modules doivent respecter des patrons de conception utilisés en Java, afin d'aider les futurs programmeurs qui souhaiteraient étendre AMELIS ou encore l'utiliser dans le cadre d'autres projets. AMELIS est composé de plusieurs modules : le cœur du programme, l'interface, le lanceur, la base de données ou persistance.

Il est par ailleurs primordial qu'AMELIS soit codé correctement, respecte les normes usuelles de programmation en Java et soit compréhensible par des personnes externes au projet.

## 2.2 Objectifs liés à PRADHA

### 2.2.1 Objectifs au niveau de la programmation

PRADHA est le programme servant à identifier les activités réalisées dans l'appartement. Il doit d'une part être capable de récupérer les données provenant des capteurs de l'appartement et d'autre part effectuer un traitement sur ces données pour déduire l'activité qui a été réalisée.

De la même façon qu'AMELIS, PRADHA est codé en Java, et on souhaite que ce programme soit modulaire. La modularité du programme doit permettre à d'autres personnes ayant besoin de faire de la reconnaissance d'activité d'utiliser les algorithmes déjà implantés et de récupérer leurs résultats en fonction des données fournies en entrée. De plus, un module de communication avec AMELIS doit permettre de séparer l'algorithme de reconnaissance de la détermination et l'affichage de l'activité à reconnaître. La décomposition de PRADHA en modules est alors essentielle au bon fonctionnement du programme en collaboration avec AMELIS et les capteurs de l'appartement.

## 2.2. OBJECTIFS LIÉS À PRADHA

### 2.2.2 Objectifs au niveau de la reconnaissance

On souhaite principalement avec PRADHA réaliser un programme de reconnaissance d'activité avec, comme dit en section 1.3, un algorithme utilisant une méthode d'apprentissage automatique. Par-ailleurs, un certain nombre de contraintes du problème nous obligent à formuler plusieurs objectifs au niveau de la reconnaissance.

**Reconnaissance de haut niveau :** On désire avoir un haut taux de reconnaissance des activités, car une activité réalisée qui n'a pas été reconnue sera rappelée, ce qui peut perturber l'utilisateur. De plus, les activités à reconnaître doivent être des activités de haut-niveau, c'est-à-dire des activités peu précises, dont on ne souhaite pas connaître les détails de réalisation. Par-exemple, on veut savoir si la personne a préparé à manger et non si elle a préparé une salade ou des spaghettis. Les activités à reconnaître dans l'appartement sont principalement des AVQ, et on ne souhaite pas faire de suivi à la réalisation de ces activités ; c'est pourquoi il n'est pas utile de reconnaître comment l'activité est réalisée.

**Reconnaissance généraliste :** La reconnaissance d'activité doit pouvoir fonctionner pour n'importe quelle personne réalisant l'activité dans l'appartement. Le système ne doit donc pas être propre à une personne mais portable pour fonctionner avec n'importe qui. Ceci afin d'être capable de porter le système sur plusieurs appartements à la fois, du moment qu'ils ont la même configuration, avec n'importe quel utilisateur.

**Respect de la vie privée :** La reconnaissance d'activité doit se faire de façon non-intrusive. C'est pourquoi l'utilisation de caméras ou de tout autre moyen d'enregistrement du son ou de l'image est banni. La reconnaissance doit donc utiliser uniquement les informations provenant de capteurs.

**Reconnaissance d'un utilisateur seul :** Pour pouvoir reconnaître dans un appartement les activités réalisées par plusieurs personnes en même temps, il est primordial de pouvoir identifier et différencier ces personnes, ce que seuls permettent les caméras et les tags RFID. Or, on ne souhaite pas utiliser d'étiquettes RFID car ces dernières sont peu pratiques puisqu'elles doivent être portées constamment par l'utilisateur. Il n'est

## 2.3. MÉTHODOLOGIE

pas possible d'utiliser de caméras pour les raisons citées plus haut. La reconnaissance ne peut donc pas fonctionner lorsque plusieurs personnes sont dans l'appartement.

**Reconnaissance quasi temps-réel :** Il faut que l'algorithme de reconnaissance d'activité soit suffisamment performant pour reconnaître l'activité pratiquement au moment où la personne réalise l'activité. En effet, si le programme prend plusieurs minutes à identifier l'activité, le système risque de réagir trop tard pour assister correctement la personne.

### 2.2.3 Objectifs au niveau de l'utilisation d'AMELIS

En plus de créer un calendrier capable de reconnaître les activités dans l'appartement, un des objectifs de la maîtrise se situe au niveau de l'utilisation qu'une personne peut faire du système AMELIS-PRADHA. Il est donc important de savoir si ce système est utile, s'il a une influence sur la réalisation des AVQ et s'il est apprécié par l'utilisateur. Le but est donc de répondre à certaines questions, notamment : est-ce que les activités rappelées par le systèmes sont réalisées en priorité? Est-ce que l'utilisateur apprécie qu'on lui rappelle plusieurs fois une activité qu'il n'a pas réalisée? Est-ce que le système aide l'utilisateur à organiser sa journée? Est-ce qu'il trouve une utilité à un tel système? Cette liste sera complétée au chapitre 7.

## 2.3 Méthodologie

Nous établirons dans cette partie la méthodologie suivie afin d'atteindre les objectifs énoncés précédemment.

### 2.3.1 Réalisation du calendrier AMELIS

AMELIS est réalisé en équipe avec Yannick Adeline, étudiant en doctorat à DOMUS travaillant sur la prise en main d'outils technologiques destinés aux personnes âgées atteintes de démence. Il est réalisé en suivant la méthode de programmation XP [2], méthode demandant aux participants du projet de travailler par paire, sur un même ordinateur : un qui code et l'autre qui vérifie que le code est bien en accord

## 2.3. MÉTHODOLOGIE

avec ce qui est demandé pour le projet. Cette méthode favorise les échanges entre les programmeurs, car le coéquipier « vérificateur » doit remettre en question chaque ligne codée, s’interroger sur sa lisibilité et sa pertinence. Les deux coéquipiers doivent aussi réfléchir ensemble sur le modèle de représentation, les patrons de conception à appliquer et la méthode de sauvegarde des données à utiliser.

Comme tous les programmes Java réalisés au DOMUS, AMELIS a été conçu pour fonctionner avec le système de gestion et d’automatisation de production de logiciels Java Maven<sup>1</sup>. Ce logiciel permet de gérer automatiquement les dépendances que pourrait avoir un logiciel produit par le laboratoire. Les dépendances peuvent ou bien être des bibliothèques présentes dans les dépôts de Maven, donc des bibliothèques utilisables par tous les développeurs Java, ou bien des bibliothèques créées au sein du laboratoire, donc utilisables uniquement par les membres de DOMUS. Ces dépendances sont téléchargées à la première compilation du programme et permettent de produire le logiciel attendu à partir des sources, tout en évitant les problèmes d’incompatibilité de mise à jour des logiciels.

Un logiciel d’intégration continue, Jenkins<sup>2</sup>, est utilisé par le laboratoire afin de générer les projets sous forme de .jar, directement utilisables par les utilisateurs. Il permet également d’exécuter les tests unitaires codés pour le projet afin de vérifier que les mises à jour effectuées sur le projet n’affectent pas le fonctionnement de base prévu pour le programme. Grâce à Jenkins, il est possible de générer des bibliothèques utilisables par les autres étudiants du laboratoire, ce qui répond au deuxième objectif d’AMELIS.

### 2.3.2 Réalisation de PRADHA

La programmation de PRADHA repose sur les mêmes principes que ceux d’AMELIS : l’utilisation de Maven et Jenkins, permettant notamment aux algorithmes d’être implantés en tant que bibliothèques utilisables par n’importe quel autre programme de DOMUS. Ce principe permet également de facilement récupérer les événements des capteurs de l’appartement grâce à l’implantation du service OSGi<sup>3</sup> faite dans l’ap-

---

1. [www.maven.apache.org](http://www.maven.apache.org)
2. [www.jenkins-ci.org](http://www.jenkins-ci.org)
3. [www.osgi.org](http://www.osgi.org)



## 2.3. MÉTHODOLOGIE

partement DOMUS par Charles Goin-Vallerand dans le cadre d'une maîtrise au laboratoire [18, 19].

L'étape la plus importante de la réalisation de PRADHA consiste en l'implantation du programme de reconnaissance d'activité. Pour répondre aux besoins du problème, le programme doit utiliser une méthode d'apprentissage automatique à partir de jeux de données issus d'activités réalisées par le passé dans le laboratoire. Ce modèle d'activité est ensuite utilisé pour identifier une activité réalisée dans l'appartement. Pour réaliser le programme de reconnaissance d'activité, la méthodologie est la suivante :

1. Répertorier les algorithmes de classification de données avec apprentissage qui répondent aux critères de sélection.
2. Implanter les algorithmes qui paraissent les plus pertinents, soit deux ou trois.
3. Tester ces algorithmes sur un jeu de données différent du jeu utilisé pour l'apprentissage.
4. Comparer les résultats des algorithmes.
5. Sélectionner l'algorithme qui donne les meilleurs résultats.
6. Interfacer le programme de classification avec les capteurs de l'appartement.
7. Interfacer le programme avec AMELIS.
8. Vérifier que le programme fonctionne dans l'appartement en quasi temps-réel.

### 2.3.3 Validation de AMELIS-PRADHA

Une fois les étapes précédentes achevées, on est en mesure de faire de la reconnaissance d'activité à partir de données fournies par les capteurs et ainsi savoir si la personne a réalisé ou non l'activité prévue, et lui rappeler à nouveau si nécessaire ; on ne connaît cependant pas l'impact que peut avoir un tel système sur l'utilisateur dans l'organisation de sa journée. C'est pourquoi il est nécessaire de réaliser des expérimentations dans un appartement, simulant les conditions réelles dans lesquelles un utilisateur pourrait se retrouver : une quinzaine d'AVQ à réaliser sont programmées dans AMELIS et PRADHA se charge de vérifier qu'elles ont bien été réalisées. Si elles ne sont pas réalisées, AMELIS les rappelle à nouveau. Ceci permet de vérifier l'impact que peut avoir un tel rappel sur l'utilisateur dans la réalisation de ses AVQ.

## 2.4. DESCRIPTION DE L'APPARTEMENT DU LABORATOIRE

L'expérimentation doit aussi permettre, via un questionnaire donné à remplir à la fin par l'utilisateur, de savoir s'il est satisfait d'un tel système et dans quelles situations il lui semble qu'il serait utile.

Une telle expérimentation doit aussi permettre de vérifier que le programme de reconnaissance d'activité fonctionne dans des conditions réelles de vie quotidienne, en mesurant le nombre d'activités reconnues et non reconnues, permettant ainsi de valider le taux de reconnaissance.

La méthodologie générale à suivre pour créer un calendrier électronique capable de reconnaître les activités réalisées dans l'appartement et évaluer son utilisation par une personne réelle est la suivante :

- implanter le calendrier électronique AMELIS
- implanter le programme de reconnaissance d'activité PRADHA
- faire fonctionner AMELIS et PRADHA ensemble
- vérifier que le duo AMELIS-PRADHA fonctionne correctement grâce à une expérimentation dans l'appartement.
- analyser les données de l'expérimentation comprenant le taux de reconnaissance et la satisfaction de l'utilisateur.

## 2.4 Description de l'appartement du laboratoire

### 2.4.1 Présentation

Le laboratoire DOMUS est un laboratoire inter-disciplinaire dédié à la domotique et à l'informatique mobile à l'Université de Sherbrooke<sup>4</sup>. Les infrastructures de ce laboratoire consistent en un appartement de taille standard, entièrement équipé, avec une cuisine, salle à manger, salon, chambre et salle de bain. Cet appartement est équipé de nombreux capteurs de type infrarouge (CI), contacteur de porte (CP), capteur d'interrupteur de lumière (CL), débitmètre (CD) (figure 2.1). S'ajoutent à ceux-ci des tapis tactiles, des repères de tiroirs et placards, des capteurs de puissance électrique, des microphones, des écrans et des effecteurs ; les éléments de cette dernière liste ne sont pas représentés sur la figure afin de l'alléger.

---

4. [www.domus.usherbrooke.ca](http://www.domus.usherbrooke.ca)

## 2.4. DESCRIPTION DE L'APPARTEMENT DU LABORATOIRE

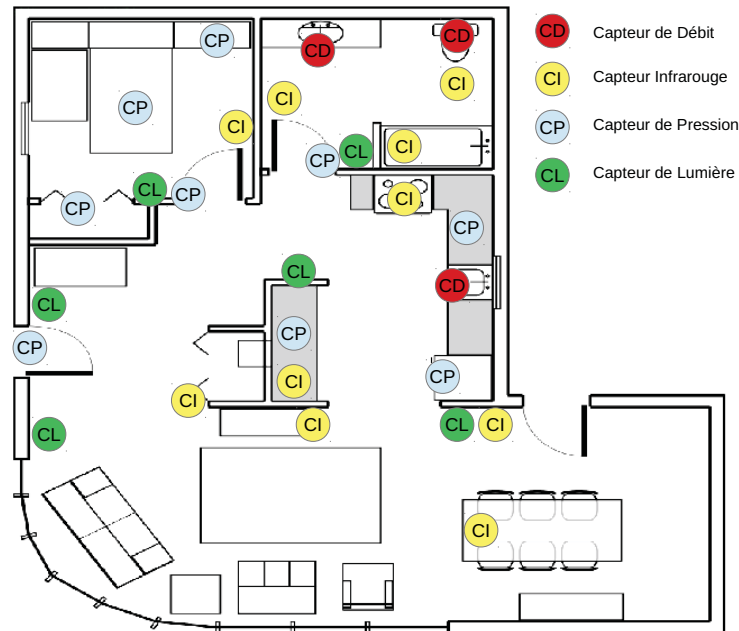


figure 2.1 – Schéma de l'appartement du laboratoire DOMUS montrant la disposition des capteurs

Ces capteurs sont utilisés dans de nombreux projets réalisés au sein du laboratoire comme la localisation [40], l'assistance à la réalisation de recettes [16], la reconnaissance de plan [4], la reconnaissance d'activité [9], l'identification d'appareils ou de personnes [8] *etc.* Le laboratoire DOMUS offre un environnement permettant de réaliser des recherches sur les environnements pervasifs [17] et l'informatique diffuse [16].

L'infrastructure du laboratoire servira dans notre cas à vérifier que la reconnaissance d'activité peut fonctionner dans un véritable appartement, avec des personnes mises en situation réelle.

### 2.4.2 Équipement

Les capteurs et systèmes d'enregistrement présents dans le laboratoire sont de différents types.

**Capteurs infrarouges :** Ce sont les capteurs les plus courants. Ils permettent de

## 2.4. DESCRIPTION DE L'APPARTEMENT DU LABORATOIRE

détecter la présence de personnes dans une pièce. Ils fournissent donc une information fiable sur la présence ou non d'une personne, et fonctionnent avec ou sans luminosité dans la pièce.

**Contacteurs ou capteurs de pression :** Ils sont placés sur les portes des pièces, sur les portes de placards ainsi que les tiroirs. On obtient grâce à eux une information sur l'ouverture et la fermeture de portes. Cependant, cette information n'est pas toujours fiable car elle dépend du soin avec lequel la personne referme les portes.

**Débitmètres :** Ils fournissent une information sur l'ouverture et la fermeture de robinets. Ils différencient l'eau chaude de l'eau froide et on les retrouve dans la salle de bain et la cuisine.

**Capteurs d'interrupteur des lumières :** Ils indiquent si une lumière précise est allumée. Ils permettent donc de savoir si une personne entre dans une pièce, bien que cette information ne soit pas totalement fiable puisqu'une personne peut entrer dans une pièce sans allumer la lumière.

**Tapis tactiles :** Ils fournissent une information sur la présence d'une personne à un endroit précis. Ils sont cependant contraignants au niveau de l'installation et de l'entretien (pour faire le ménage) et sont peu esthétiques s'ils ne sont pas cachés sous un véritable tapis. Ils n'ont donc pas été utilisés dans le cadre de cette maîtrise.

**Appareils de prise de vidéo :** Une caméra est située sur le plafond, dans le salon avec une tête orientable. Elle sert à analyser ce que font les utilisateurs.

**Microphones :** Des micros sont dispersés dans chaque pièce de l'appartement.

**Capteurs Watteco :** Des capteurs sont placés sur les panneaux électriques permettant d'enregistrer la puissance électrique consommée par les appareils branchés sur chacune des phases.

Tous ces capteurs ont été utilisés dans le cadre de cette maîtrise pour réaliser les expérimentations :

- Les capteurs infrarouges, de pression, les débitmètres et les capteurs d'interrupteur des lumières sont utilisés par PRADHA pour reconnaître les activités

## 2.4. DESCRIPTION DE L'APPARTEMENT DU LABORATOIRE

- réalisées dans l'appartement.
- Les tapis tactiles, placés devant le calendrier AMELIS sont utilisés pour savoir si le participant consulte le calendrier.
  - Les appareils de prise de vidéo et de son sont utilisés pour vérifier ce que l'utilisateur est en train de faire et ainsi s'assurer que le programme de reconnaissance d'activité fonctionne bien. D'autre part, la prise de son et de vidéo permet à d'autres étudiants du laboratoire intéressés par les données des capteurs de les annoter afin de les utiliser dans le cadre d'un algorithme d'apprentissage d'activité.
  - Les capteurs Watteco sont utilisés pour générer un nouveau jeu de données de données électriques pouvant servir à de futures expérimentations physiques impliquant les consommations électriques des appareils.

L'installation de DOMUS permet d'utiliser le réseau pour échanger des données entre les appareils, les capteurs et les serveurs par-exemple, et peut fournir en électricité certains de ces appareils, tels que les systèmes d'éclairage et les micros.

L'ensemble des capteurs utilisés pour faire la reconnaissance d'activité uniquement est répertorié dans le tableau 2.1.

Type de capteur	Nombre de capteurs				
	Cuisine	Salle de bain	Chambre	Entrée	Salon / salle à manger
Infrarouge	2	3	1	1	3
Contacteur pour placard	27	4	10	2	9
Contacteur pour porte de pièce	0	1	1	1	0
Débitmètre	2	5	0	0	0

tableau 2.1 – Répartition des différents capteurs dans l'appartement utilisés par PRADHA

# Chapitre 3

## AMELIS, un calendrier électronique adapté aux personnes âgées

La première étape à suivre, d'après la méthodologie expliquée au chapitre 2, consiste à créer et implanter le calendrier AMELIS, en Java. Dans ce chapitre, nous allons voir comment cette implantation a été faite, les difficultés rencontrées et les choix de représentation faits.

### 3.1 Introduction

#### 3.1.1 Présentation

Le calendrier AMELIS (Alzheimer Mémoire et LIens Sociaux) est un projet du laboratoire qui a été initié par Céline Descheneaux et Hélène Pigot en 2009 [14].

AMELIS est une innovation qui est bâtie à partir des besoins des personnes âgées :

- de se rappeler des événements futurs à planifier,
- de se remémorer des événements passés,
- de créer des liens sociaux en partageant les fêtes familiales,
- de suivre les rituels liés aux grandes fêtes officielles.

### 3.1. INTRODUCTION

AMELIS a été conçu en s'appuyant sur les capacités des personnes âgées à récupérer de l'information sur un écran et à utiliser des calendriers muraux.

Le calendrier AMELIS poursuit donc un triple but :

- aider à l'organisation de la vie quotidienne en affichant de façon interactive les rendez-vous et les tâches importantes à se rappeler
- aider à l'orientation temporelle, pour le présent, la mémoire des faits passés et ceux à venir
- aider au maintien des contacts interpersonnels.

Le calendrier prend en compte plusieurs aspects et connaissances à propos de l'information du domicile, de la conception centrée utilisateur et des besoins des personnes âgées.

Dans le cadre de cette maîtrise, l'accent est mis sur l'aide à l'organisation de la vie quotidienne. Les autres buts sont mis de côté pour d'autres projets qui pourront servir à compléter AMELIS.

#### 3.1.2 Interface



figure 3.1 – Capture de l'écran d'accueil d'AMELIS

L'interface d'accueil d'AMELIS (figure 3.1) est divisée en deux parties :

**Le calendrier :** il est situé dans la partie droite de l'écran. Il peut être sous la forme d'un calendrier ou bien mensuel, ou bien hebdomadaire. Des icônes dans les cases

### 3.1. INTRODUCTION

des jours permettent d'identifier rapidement les activités spéciales telle que les prises de médicaments, les anniversaires ou les fêtes nationales. Il est possible de naviguer dans les semaines ou les mois à l'aide de flèches. L'utilisateur peut sélectionner une case pour avoir des détails sur les activités de la journée. Ces détails s'affichent dans le panneau de gauche.

**Le panneau de gauche :** il est situé dans la partie gauche de l'écran. L'écran d'accueil est une photographie, choisie par l'utilisateur. Le haut du panneau sert à orienter temporellement l'utilisateur en précisant la date, l'heure, si c'est le matin ou le soir ainsi que le temps dehors. Le bas du panneau est un panneau de contrôle qui permet de gérer les activités enregistrées dans le calendrier. C'est principalement avec ce panneau que l'utilisateur va interagir, notamment pour ajouter et planifier ses activités.

Lorsque l'utilisateur sélectionne un jour de la semaine sur le panneau de droite, il obtient le détail de la journée dans le panneau de gauche (figure 3.2).



figure 3.2 – Capture de l'écran de détail de la journée

#### 3.1.3 Fonctionnalités

Pour répondre aux buts énoncés précédemment, un certain nombre de fonctionnalités doivent être implantées.



### 3.1. INTRODUCTION

**Activités prédéfinies** Un des défis important d'AMELIS est de permettre aux personnes âgées de l'utiliser à bon escient, au maximum de ses capacités. Cependant, il est difficile pour une personne âgée d'entrer de nombreuses informations, de multiplier les étapes nécessaires à l'enregistrement d'une activité dans le calendrier. Il faut la plupart du temps qu'elle soit aidée par un proche qui puisse entrer les informations.

Afin de permettre à une personne âgée de facilement planifier ses activités, il a été décidé de permettre aux proches d'aider les personnes âgées en pré-enregistrant les activités à planifier. Des activités prédéfinies ont donc été mises en place ; il s'agit d'activités contenant toutes les informations concernant le lieu où elles doivent se réaliser, les images associées et les gens qui participent, mais pour lesquelles n'est associée aucune date. C'est lorsque la personne va planifier une de ces activités prédéfinies qu'elle va lui assigner une ou plusieurs dates. On peut donc voir une activité comme étant une instance d'une activité prédéfinie à un moment précis. La tâche de planification d'activité est allégée pour la personne âgée puisqu'elle n'a plus qu'à sélectionner parmi une liste l'activité à planifier et lui affecter une date.

**Médicaments** Il est important lors de l'enregistrement d'activités de différencier les activités de prise de médicaments des autres activités. En effet, l'affichage est différent puisqu'une icône de médicament apparaît lorsqu'il y a une prise planifiée dans la journée, pour s'assurer que la personne voie qu'elle a un médicament à prendre ce jour-ci. De plus, on peut supposer que dans les futurs projets d'AMELIS, il y ait besoin de faire des traitements spéciaux pour la prise de médicaments, comme un rappel supplémentaire ou une alarme spécifique, ou encore la mise en place d'une récurrence spécifique pour les prises régulières.

Les activités de prises de médicaments se planifient donc à partir de médicaments pré-enregistrés dans AMELIS, et non à partir d'activités prédéfinies. Cette façon de faire permet de limiter les erreurs de saisie, car une prise de médicament manquante ou supplémentaires peut avoir des conséquences néfastes pour l'utilisateur. Ainsi, les activités de médication se gèrent comme suit : la personne âgée a à sa disposition la liste des médicaments qu'elle prend habituellement ou qu'elle a déjà pris au moins une fois, et doit planifier leur prise. Une fois cette prise planifiée, une petite icône représentant des médicaments apparaît sur le jour correspondant.

## 3.2. VERSION PARTANT DE LA BASE DE DONNÉES

**Contacts** AMELIS n'est pas qu'un simple calendrier. Il doit servir, en plus de rappeler les activités à réaliser, à aider l'utilisateur à garder contact avec ses proches. C'est pourquoi AMELIS contient un carnet d'adresses avec toutes les personnes qu'elle peut avoir besoin de contacter, avec les informations d'adresse, de numéro de téléphone, de date d'anniversaire *etc.* Ce carnet d'adresse permet également de rappeler à la personne âgée les anniversaires à souhaiter.

**Photos** L'utilisation d'images est important pour le calendrier puisque les images permettent la réminiscence des souvenirs et le rappel des événements marquants. Il faut donc offrir la possibilité à la personne d'enregistrer pour chaque activité et chaque contact des photos ou images évocatrices.

### 3.1.4 Équipement

Le logiciel est destiné à fonctionner sur un MacMini<sup>®</sup>, choisi pour sa petite taille, interfacé avec un écran tactile Dell<sup>®</sup> accroché au mur ou placé sur une petite table. Il est supposé permettre aux personnes âgées de planifier leurs activités de façon simple et efficace, avec le minimum d'étapes, tout en offrant la possibilité d'enregistrer le maximum d'informations. Pour ce faire, deux versions majeures d'AMELIS ont été implantées. Elles sont décrites dans les sections suivantes.

## 3.2 Version partant de la base de données

La première version a été conçue à partir de la représentation choisie pour stocker les données. Cela signifie que la façon d'enregistrer les données entrées dans le calendrier a été choisie en priorité, avant même de savoir comment elles seraient représentées en Java. Il faut pour cela utiliser un système de gestion de base de données (SGBD). Le SGBD choisi pour le projet est PostgreSQL<sup>1</sup>. Il s'agit d'un SGBD libre et gratuit offrant les mêmes fonctionnalités avancées que les SGBD commerciaux. Il offre notamment la possibilité d'utiliser le langage procédural PL/pgSQL qui permet

---

1. [www.postgresql.org](http://www.postgresql.org)

### 3.2. VERSION PARTANT DE LA BASE DE DONNÉES

d'automatiser certaines tâches lors de l'insertion de nouveaux éléments dans la base. Il permet également de créer des fonctions de traitements automatiques sur les données.

Les étapes suivies pour réaliser cette première version d'AMELIS sont donc les suivantes :

1. création des tables pour sauvegarder les informations que l'on estime nécessaires au bon fonctionnement d'AMELIS,
2. création du modèle Java équivalent,
3. création des classes permettant de récupérer les informations de la base de données (BD) et de les utiliser comme modèle Java,
4. création des classes permettant d'enregistrer les informations du modèle Java que l'on juge importantes dans la BD.

Il est nécessaire de créer les tables correspondant aux données que l'on souhaite sauvegarder et à partir de là de créer le modèle objet en Java, qui est supposé identique. Avec cette méthode, le but est de minimiser le nombre de tâches à effectuer en Java en les codant en priorité dans la base de données. Ainsi, les erreurs et les rappels à effectuer en fonction de l'heure courante sont gérés directement dans la base grâce à des fonctions codées à cet effet.

Le fait de partir de la BD pour créer le modèle de représentation de données Java engendre de nombreuses difficultés, car avec cette façon de faire, nous inférons que les traitements effectués sur les modèles Java sont les mêmes que sur la base de données, ce qui se révèle faux. En effet, la création du modèle à partir de la BD ne tient pas compte de l'interfaçage avec Java. Une fois la programmation d'AMELIS commencée, on se rend compte qu'il y a des incohérences dans le modèle de la BD, ou bien que certaines fonctionnalités nécessaires au bon fonctionnement du programme ne peuvent pas être implantées avec cette représentation. Il faut alors modifier la BD en conséquence, ce qui fait faire beaucoup d'aller-retours et donc perdre beaucoup de temps, pour finalement obtenir un résultat peu fonctionnel et peu souple.

De plus, le programme est sensé pouvoir fonctionner sans la BD. En effet, nous pourrions imaginer le cas où le SGBD ne serait pas mis en place sur l'ordinateur mais on souhaiterait malgré tout utiliser AMELIS, simplement pour avoir un calendrier consultable, sans avoir à enregistrer les activités. Cette alternative est impossible

### 3.3. VERSION PARTANT DU MODÈLE JAVA

avec la façon de faire précédente. Il est également impossible de changer PostgreSQL pour un autre système de sauvegarde de données, comme par exemple MySQL<sup>2</sup> ou simplement un fichier texte. Enfin, cette méthode ne permet pas d'obtenir l'indépendance des modules, donc l'indépendance entre l'interface, la BD et le cœur.

On peut retrouver la liste des avantages et inconvénients de la méthode choisie pour cette première version d'AMELIS dans le tableau 3.1.

Avantages	Inconvénients
Utilise les fonctionnalités avancées de la BD	Fort couplage avec la BD
Permet d'optimiser le traitement des données de la BD	Nécessite un grand nombre d'aller-retours
Diminue le nombre d'accès à la BD	Ne permet pas au programme de fonctionner sans la BD

tableau 3.1 – Avantages et inconvénients de la première méthode d'implantation choisie

Finalement, avec cette première version, nous avons pu obtenir un calendrier avec pratiquement toutes les fonctionnalités requises. Elle n'a cependant pas abouti, à cause de sa non-modularité, son trop fort couplage avec la BD et ses erreurs de représentation. Il a donc fallu reprendre complètement le projet, en ne gardant que les bonnes idées du projet précédent, afin d'obtenir un projet vraiment stable, modulaire et compréhensible par des personnes de l'extérieur.

### 3.3 Version partant du modèle Java

Une deuxième version d'AMELIS a vu le jour lorsqu'il a été nécessaire de mettre le projet en intégration continue, avec l'aide de Jenkins. La première version ne permettait pas cette intégration en continu, de par sa structure non-modulaire. Dans cette deuxième version, la BD n'a pas été mise en place lors de la programmation d'AMELIS, ce qui a permis de faire un programme complètement indépendant de sa méthode de sauvegarde de données.

---

2. [www.mysql.com](http://www.mysql.com)

### 3.3. VERSION PARTANT DU MODÈLE JAVA

#### 3.3.1 Architecture et fonctionnalités

Le découpage du projet AMELIS en plusieurs projets permet la décomposition en modules. Ainsi, le projet AMELIS a été découpé en trois projets principaux :

- *domus-amelis-core* : contient le cœur du programme, c'est-à-dire les entités du programme, les modèles et les contrôleurs manipulant ces modèles. Le programme doit donc être capable de fonctionner uniquement grâce à ce projet, et l'interface graphique est une couche s'ajoutant au projet permettant à l'utilisateur d'interagir avec le programme. Il n'a aucune dépendance sur d'autres projets.
- *domus-amelis-gui* : contient l'interface graphique. Le gestionnaire Java d'interface choisi est javax.swing ; cependant il doit être possible de faire fonctionner AMELIS avec un autre gestionnaire en simplement créant une autre version de *domus-amelis-gui* utilisant cet autre gestionnaire. Ce projet a pour rôle de représenter les modèles présents dans le cœur du programme, il a donc pour dépendance le projet *domus-amelis-core*.
- *domus-amelis-launcher* : ce projet est le lanceur du programme. C'est lui qui crée la fenêtre graphique avec les liens entre les modèles de *domus-amelis-core* et *domus-amelis-gui*.

Une représentation des dépendances est donnée en figure 3.3.

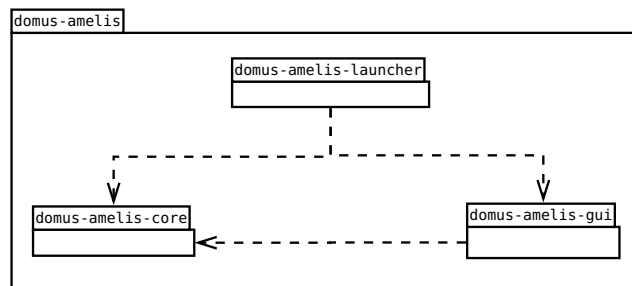


figure 3.3 – Arbre des dépendances dans AMELIS

Sur cette représentation, la BD n'apparaît pas dans l'arbre des dépendances. En effet, aucun système de persistance des données n'a été mis en place dans cette version, ce qui contraste grandement avec la première version qui se basait entièrement sur la

### 3.3. VERSION PARTANT DU MODÈLE JAVA

BD. Étant donné que la création d'un système de persistance ne paraît plus comme une priorité dans le programme, son implantation a été repoussée à plus tard. Dans ce cas, on peut se servir normalement du calendrier, donc planifier des activités et se les faire rappeler, ajouter des contacts et des médications. Cependant, à chaque fois que le programme sera redémarré, toutes les données ajoutées au calendrier seront perdues, car elles ne sont pas persistantes. Ceci ne permet pas une utilisation du calendrier dans de réelles conditions, chez des personnes, mais offre la possibilité de faire des expérimentations de courte durée, pour faire des tests ciblés sur certaines parties du programme.

Le découpage choisi permet d'utiliser un patron de conception couramment utilisé en Java, surtout pour les programmes comportant une interface graphique : le patron Modèle-Vue-Contrôleur (MVC). Il permet de séparer l'interface du cœur du programme, autorisant ainsi le changement du gestionnaire de fenêtres par exemple sans avoir à modifier le cœur du programme. En annexe A se trouvent deux diagrammes de classes illustrant l'application du modèle MVC dans la programmation d'AMELIS.

#### 3.3.2 Représentation d'une activité

Dans cette version a été appliquée la notion d'activité prédéfinie telle qu'il a été annoncé en section 3.1.3. Le diagramme de classes présenté en figure 3.4 détaille la relation existant entre les activités prédéfinies et les activités planifiées, incluant les activités de prise de médicaments.

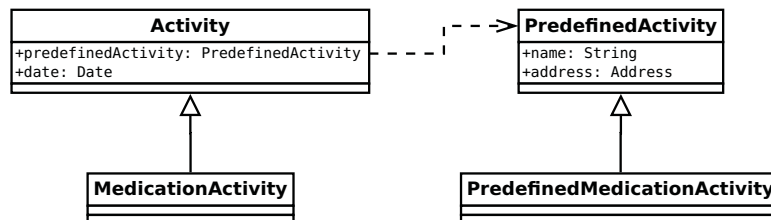


figure 3.4 – Représentation des entités principales d'AMELIS

## 3.4. BILAN

### 3.3.3 Interface d'AMELIS

La réalisation du calendrier AMELIS nécessite d'avoir des connaissances d'une part en programmation, et d'autre part en design d'interfaces. Le design d'interfaces ne faisant pas partie du domaine de l'informatique, il a fallu faire appel à des designers afin qu'ils proposent des interfaces d'AMELIS correspondant à nos besoins.

Des designers ont donc proposé diverses interfaces répondant à certains critères de clarté et d'esthétisme. De plus, les interfaces devaient être faites de telle sorte à ce qu'elles permettent aux personnes âgées atteintes de démence d'utiliser facilement le calendrier. Les paramètres jouant sur l'utilisabilité d'un tel calendrier sont multiples : emplacement des boutons, couleurs, contrastes, taille des polices, disposition générale des différents éléments *etc.*

Une fois que l'interface jugée la meilleure a été sélectionnée, il a fallu utiliser les fonctionnalités offertes par `javax.swing`, le gestionnaire de fenêtres, pour faire correspondre autant que possible l'interface créée par les designers avec celle du programme. L'outil de design offert par Netbeans<sup>®</sup> a permis de créer les interfaces voulues facilement.

## 3.4 Bilan

La programmation d'AMELIS s'est faite en deux étapes principales : la première, qui n'a pas abouti, a permis de connaître les difficultés et les erreurs à éviter en programmation. On a pu grâce à cette première version créer une deuxième version stable, avec un bon modèle de représentation de données et une bonne décomposition modulaire.

Cependant, certaines fonctionnalités indispensables sont encore manquantes dans cette deuxième version, notamment la persistance des données. Il est donc prévu de mettre en place prochainement un SGBD via le *framework* Hibernate<sup>3</sup>. Ce *framework* permet de s'abstraire des méthodes d'accès à la BD en les remplaçant par des appels à des méthodes objet de haut niveau, directement dans le code Java.

AMELIS est donc arrivé à un stade où il est utilisable par une personne âgée, bien

---

3. [www.hibernate.org](http://www.hibernate.org)

### 3.4. BILAN

que les données ne soient sauvegardées dans aucune BD. Les fonctionnalités de base d'AMELIS comme la saisie de nouvelles activités et le rappel de ces activités sont implantées ; ceci permet de passer à l'étape suivante du projet de maîtrise qu'est l'immersion d'AMELIS dans l'environnement intelligent afin de vérifier que les activités planifiées ont bien été réalisées.



# Chapitre 4

## Pradha, logiciel de reconnaissance d'activité dans un appartement intelligent

Le logiciel de reconnaissance d'activité PRADHA est couplé au calendrier AMELIS pour lui permettre de savoir si une activité inscrite à l'agenda a été réalisée. Dans ce chapitre, nous présenterons les objectifs, puis les jeux de données qui ont aidé à la sélection des algorithmes. Ensuite seront décrits les algorithmes choisis et implantés pour faire la reconnaissance d'activité.

### 4.1 Introduction

#### 4.1.1 Objectifs du projet PRADHA

Tel qu'énoncés au chapitre 2, les objectifs au niveau du projet PRADHA sont multiples. Il s'agit de la deuxième partie de notre projet de maîtrise, venant après le calendrier électronique. Cette partie doit permettre, à partir d'une connaissance *a priori* de l'activité à réaliser, de l'identifier et ainsi vérifier qu'elle a bien été réalisée. L'objectif principal est donc de concevoir un programme de reconnaissance d'activité à partir de données de capteurs. Pour ce faire, il faut créer des modèles d'activité à l'aide d'une méthode d'apprentissage automatique. On a vu à la section 2.3.2 qu'un

## 4.1. INTRODUCTION

Le jeu de données disponible au DOMUS permettrait de faire l'entraînement de notre algorithme de reconnaissance d'activité. Ce jeu de données a été généré à partir de l'appartement du laboratoire, où a lieu la reconnaissance d'activité, c'est-à-dire où est installé AMELIS. C'est donc à partir de ce jeu de données qu'on a pu établir nos modèles d'activité et vérifier le taux de reconnaissance des algorithmes choisis.

Pour créer un modèle d'activité, nous identifions les algorithmes qui s'appliquent à notre problème afin de sélectionner les plus efficaces. La démarche consiste donc, à partir de la revue de littérature présentée en 1.3, à identifier les algorithmes qui pourraient répondre à notre problème. Une fois qu'on a sélectionné un algorithme, on l'applique sur le jeu de données fourni par le DOMUS pour faire l'apprentissage. Ceci fait, on vérifie qu'il identifie correctement les activités réalisées à partir d'activités provenant d'un autre jeu de données.

### 4.1.2 Jeu de données

#### Génération du jeu de données

Le jeu de données utilisé pour entraîner et valider l'algorithme de reconnaissance d'activité est un jeu de données généré en 2008 pendant une expérimentation au laboratoire [9] où des personnes ont dû :

- se réveiller,
- aller aux toilettes,
- préparer le déjeuner,
- manger,
- faire la vaisselle.

L'expérimentation a été réalisée avec six personnes différentes et a généré 433 instances d'activités. Une instance correspond à une réalisation, par une personne, d'une activité particulière. Les instances recueillies ont été divisées en deux séries. La première série correspond à la première fois où les personnes ont fait l'expérimentation et a généré 307 instances d'activité. La seconde série correspond à la seconde fois où les personnes ont dû faire l'expérimentation et a généré 126 instances d'activité. Toutes ces données sont téléchargeables directement sur le site web du laboratoire :

## 4.1. INTRODUCTION

<http://domus.usherbrooke.ca/jeux-de-donnees>.

La deuxième série a été utilisée pour l'apprentissage car 126 instances réparties sur cinq activités différentes est largement suffisant pour faire un apprentissage. Il y a eu entre 20 et 31 instances d'activités, avec une moyenne de 25.2 instances par activité, ce qui donne une bonne dispersion.

### Description du jeu de données

Le jeu de données permettant d'entraîner l'algorithme est composé de deux fichiers : le premier est généré automatiquement par le serveur de l'appartement et contient tous les événements des capteurs générés dans l'appartement avec l'heure précise de leur déclenchement. Le second est un fichier d'annotations, rempli « à la main » par une personne qui a analysé les activités réalisées dans l'appartement et a indiqué pour chacune d'elle son heure de début et de fin.

Par-exemple, d'après le fichier d'annotations, l'activité 1, qui correspond à l'activité de se réveiller, a été réalisée entre 09:17:46 et 09:18:00 (4.1).

Start time	End time	ID
11-Aug-2008 09:17:46	11-Aug-2008 09:18:00	1
11-Aug-2008 09:18:04	11-Aug-2008 09:18:51	2

figure 4.1 – Exemple d'un fichier d'annotations

En se référant au deuxième fichier, le fichier d'évènements des capteurs (figure 4.2), on peut savoir quels évènements ont été générés lors de la réalisation de cette activité, soit les capteurs 5105, 0215, 5101, 5105 et 0215.

Ainsi, on est en mesure de dire que la génération des évènements 5105, 0215 et 5101 correspond à l'activité de se réveiller. Grâce à cette connaissance, il est possible d'entraîner notre algorithme de reconnaissance d'activité. Les prochaines sections décrivent les algorithmes de reconnaissance d'activité choisis pour répondre à notre problème.

## 4.2. ALGORITHME DE SIMPLE ASSOCIATION

Date	Time	ID	Type	Room	State
2008-08-11	9:17:46	5105	Lampe	Bedroom	Open
2008-08-11	9:17:48	0215	Door	Bedroom	Open
2008-08-11	9:17:53	5101	Lampe	Kitchen	Open
2008-08-11	9:17:57	5105	Lampe	Bedroom	Close
2008-08-11	9:18:0	0215	Door	Bedroom	Close
2008-08-11	9:18:4	0216	Door	Bathroom	Open
2008-08-11	9:18:5	5106	Lampe	Bathroom	Open
2008-08-11	9:18:9	0216	Door	Bathroom	Close
2008-08-11	9:18:12	F101	Water	Bathroom	Open

figure 4.2 – Exemple d’un fichier d’évènements de capteurs

## 4.2 Algorithme de simple association

### 4.2.1 Apprentissage

Le premier algorithme choisi pour reconnaître les activités dans l’appartement est un algorithme de simple association. Il s’agit d’un algorithme simple et instinctif dont la phase d’apprentissage se déroule en plusieurs étapes pour reconnaître une activité  $A$ .

**Étape 1:** Établir un seuil  $s$  qui corresponde au seuil à dépasser pour qu’un évènement soit considéré comme un évènement clé.

**Étape 2:** Récupérer tous les évènements émis dans l’appartement pour une activité  $A$ .

**Étape 3:** Déterminer les évènements clés, c’est-à-dire les évènements générés un grand nombre de fois pour chaque activité. Pour cela, plusieurs étapes:

1. Ajouter tous les évènements au-dessus du seuil  $s$  à la liste  $L$ .
2. Assigner la liste  $L$  à l’activité  $A$ .

**Étape 4:** Répéter à partir de l’étape 2 jusqu’à ce que toutes les activités soient traitées.

Par-exemple, si on considère l’activité  $A$  « se brosser les dents », par une analyse successive des instances de cette activité, on obtient la liste d’évènements suivante:

## 4.2. ALGORITHME DE SIMPLE ASSOCIATION

{IR01:5, F105:56, IR05:0, IM02:4, 5105:28, 0210:0}. En additionnant tous ces évènements, on obtient 93 évènements générés pour l'activité  $A$ . Étant donné qu'on a établi un seuil  $s$  égal à 10%, cela signifie que les capteurs devront être activés au moins 9,3 fois pour être considérés comme capteurs clés. Donc seuls les capteurs F105 et 5105 seront des capteurs clés; on obtient alors la liste  $L : \{F105, 5015\}$ . Le modèle d'activité contiendra donc les capteurs F105 et 5105 pour l'activité « se brosser les dents ».

### 4.2.2 Validation

Pour reconnaître une activité, il faut récupérer tous les évènements générés dans l'appartement pendant un certain intervalle de temps, puis vérifier si tous les capteurs clés ont été activés. Si oui, l'activité est considérée comme réalisée; si aucun ou seulement une partie de ces capteurs clés a été activé, alors l'activité est considérée comme non réalisée.

Cet algorithme est basique, mais s'est avéré relativement efficace, en fonction des activités à reconnaître. Les résultats de validation sont présentés dans le tableau 4.1. Ils ont été établis avec un seuil  $s$  de 10%, qui est le seuil qui donne les meilleurs résultats.

Activité	Nombre total d'instances	Nombre d'instances reconnues	Pourcentage de reconnaissance
Se réveiller	47	47	100
Aller aux toilettes	66	66	100
Préparer le déjeuner	65	11	16
Manger	62	21	33
Faire la vaisselle	67	44	65
Total	307	189	61.6

tableau 4.1 – Résultats de validation de l'algorithme de simple association

On peut remarquer que les activités simples comme « se réveiller » et « aller aux toilettes » sont les mieux reconnues. Cependant, dès que les activités deviennent un peu plus complexes, ou si elles situent dans la même pièce, les résultats de reconnaissance ne sont plus aussi satisfaisants. En particulier, l'activité de préparer le déjeuner

## 4.2. ALGORITHME DE SIMPLE ASSOCIATION

est très peu reconnue. Ce résultat est dû au fait que pour préparer un déjeuner, les capteurs clés ne sont pas tellement différents de faire la vaisselle: capteur de présence dans la cuisine, capteur d'ouverture de tiroir, d'eau qui coule *etc.* L'activité est donc souvent confondue avec une autre, d'où le faible taux de reconnaissance.

### 4.2.3 Limites

Pourquoi cet algorithme ne suffit pas à avoir des résultats satisfaisants? Parce que ce qu'on appelle « capteur clé » ne correspond pas exactement à un capteur clé. Il correspond aux capteurs les plus souvent activés. Or, un capteur infrarouge sera beaucoup plus souvent activé qu'un capteur de placard, alors que le capteur de placard peut très bien être un capteur clé. Par-exemple, pour préparer le déjeuner, la personne va activer très souvent le capteur infrarouge de la cuisine, ce qui va le catégoriser automatiquement comme capteur clé. Cependant, pour préparer le déjeuner, il est indispensable de récupérer le lait ou le beurre dans le réfrigérateur. Le capteur d'ouverture du réfrigérateur devrait donc aussi être considéré comme capteur clé, alors qu'il est activé bien moins souvent que le capteur infrarouge. Le capteur du réfrigérateur risque donc à tort de ne pas être considéré comme un capteur clé par l'algorithme.

Une solution à ce problème serait d'affecter des poids à chaque capteur, mais dans ce cas on perdrait la notion d'apprentissage automatique, d'autant plus que ces poids peuvent varier d'une activité à l'autre. De plus, pour des activités proches ou similaires, l'algorithme ne pourra jamais donner de résultats fiables, étant donné que le modèle de données ne se limite qu'à certains capteurs, sans tenir compte de leur fréquence d'activation. L'algorithme est trop simple, trop naïf pour pouvoir donner des résultats satisfaisants, surtout pour des activités complexes qui demandent plus de précision que de simplement savoir si la pièce dans laquelle se trouve la personne permet de réaliser l'activité. Il faut donc se référer à ce qui a été vu dans la revue de littérature afin d'implanter un algorithme plus complexe mais plus efficace.

## 4.3 Algorithme de mixture de Gaussiennes

Afin d'expliquer le fonctionnement de l'algorithme choisi, qui est un algorithme utilisant des mixtures de Gaussiennes, nous commencerons par une présentation générale, puis expliquerons le modèle de représentation des données choisi. Suivra le détail des phases d'apprentissage et de validation.

### 4.3.1 Présentation générale

Comme expliqué en section 2.2, un des objectifs de PRADHA est de reconnaître les AVQ de haut-niveau, dont on ne souhaite pas connaître les détails. Ainsi, pour reconnaître cette activité, il faut que son modèle d'apprentissage soit un modèle global, comprenant les différentes variations que pourrait avoir cette activité.

Par-exemple, l'activité « préparer le déjeuner » est une activité très vague, qui peut avoir différentes formes en fonction de ce que prépare la personne. La personne pourrait préparer des œufs avec du pain, ou bien simplement un bol de céréales. Si elle décide de faire des œufs, elle devra utiliser des plaques de cuisson, donc activer les capteurs de la cuisinière. Elle activera en plus le capteur du réfrigérateur en allant chercher les œufs, et le grille-pain en faisant griller le pain. Si par-contre elle ne prépare qu'un bol de céréales, seul le capteur du réfrigérateur sera activé. Les deux modèles de données seront donc bien différents en fonction du repas à préparer, alors que le repas sera dans les deux cas un déjeuner. Étant donné qu'on ne veut pas faire la différence entre ces deux « sous-activités », qu'on veut uniquement reconnaître l'activité « préparer le déjeuner » dans sa globalité, il faut trouver un modèle de données qui puisse contenir plusieurs sous-modèles.

L'algorithme implanté, appelé mixture de Gaussiennes, est un algorithme qui répond à ce problème. Cet algorithme est notamment utilisé pour effectuer de la reconnaissance de caractères écrits à la main [23] et de voix [42]. Dans le cas de la reconnaissance de caractères, l'algorithme est censé reconnaître un caractère écrit peu importe la personne qui l'écrit. Pour la voix, il est supposé identifier la personne qui lit, peu importe le texte lu. Il a également été utilisé pour déterminer des habitudes et des relations dans la réalisation des AVQ [31], ce qui se rapproche davantage de l'utilisation que nous pourrions en faire dans le cadre de cette maîtrise.

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

#### 4.3.2 Représentation des données

Les données d'entrée de notre algorithme, obtenues à partir du jeu de données discuté en section 4.1.2, sont des vecteurs représentant le nombre de fois que chaque capteur est activé. À chaque fois qu'une activité est réalisée dans l'appartement, on obtient une *instance* de cette activité, ce qui donne un vecteur d'activité. On représente un vecteur d'activité par  $x$  dont la dimension  $d$  correspond au nombre de capteurs présents dans l'appartement. Un grand nombre de capteurs entraîne donc une grande dimension. Pour éviter une dimension trop élevée, il faut donc sélectionner les capteurs les plus significatifs et importants, c'est-à-dire les mieux situés et les plus différenciables. Dans notre cas, nous allons nous intéresser à seulement quatorze capteurs parmi la centaine de capteurs présents dans l'appartement: les capteurs infrarouge, les capteurs de débit d'eau et certains placards.

Ci-dessous se trouve la représentation du vecteur  $x$  représentant les différents événements capteurs qui ont été activés pour réaliser l'activité, sur une plage horaire précise. Il a été réduit à une dimension 10 pour des raisons d'espace:

$$x = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 3 \\ 2 \\ 5 \\ 0 \\ 0 \\ 4 \\ 0 \end{pmatrix} \text{ ce qui signifie: } \begin{pmatrix} \text{capteur IR01 : activé 0 fois} \\ \text{capteur IR05 : activé 1 fois} \\ \text{capteur F101 : activé 0 fois} \\ \text{capteur F102 : activé 3 fois} \\ \text{capteur IR02 : activé 2 fois} \\ \text{capteur F103 : activé 5 fois} \\ \text{capteur MW01 : activé 0 fois} \\ \text{capteur 1111 : activé 0 fois} \\ \text{capteur TC01 : activé 4 fois} \\ \text{capteur 0216 : activé 0 fois} \end{pmatrix}$$

Il est important de noter que l'ordre des capteurs, qui permet de connaître les différents éléments qui composent une activité, doit rester fixe tout au long de l'apprentissage.



### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

#### 4.3.3 Apprentissage

##### Représentation à l'aide des mixtures de Gaussiennes dans un espace multidimensionnel

Comme dit plus haut, une activité est représentée comme un modèle statistique. Étant donné la représentation choisie pour modéliser nos activités qui consiste à compter pour chaque activité les événements générés par capteur, on approxime les valeurs obtenues par des valeurs continues. En pratique, les mixtures de Gaussiennes sont le plus souvent utilisées pour représenter ce type de modèle ([46]). Cela signifie que chaque activité est représentée par plusieurs distributions de Gaussiennes  $D_j$  ayant pour paramètres  $\theta_j(\mu_j, \Sigma_j)$ .  $\mu$  est le vecteur moyenne et  $\Sigma$  est la matrice de covariance définie positive. L'ensemble des paramètres  $\theta_j$  de ces distributions est représenté par  $\Theta$ .

La représentation d'une mixture de Gaussiennes est donnée par l'équation (4.1).

$$prob(x|\Theta) = \sum_{j=1}^K \omega_j p_j(x|\theta_j) \quad (4.1)$$

où  $K$  est le nombre de distributions de la mixture.  $\Theta = (\Sigma, \mu)$  représente les paramètres de la mixture, donc des Gaussiennes,  $\omega_j$  est le poids associé à chaque Gaussienne et  $p_j(x|\theta_j)$  la probabilité que le vecteur  $x$  ait pu être généré par la distribution  $\theta_j$ .

Soit  $\mu_j$  le vecteur moyenne:

$$\mu_j = \frac{1}{N_j} \sum_k^{N_j} x_k \quad (4.2)$$

Soit  $\Sigma_j$  la matrice de covariance:

$$\Sigma_j = \frac{1}{N_j} \sum_k^{N_j} (x_k - \mu_j)(x_k - \mu_j)^T \quad (4.3)$$

Selon [30], une Gaussienne est représentée dans un espace multidimensionnel par:

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

$$p(x|\theta_j) = (2\pi)^{-d/2} |\Sigma_j|^{-1/2} \exp[-1/2(x - \mu_j)^T \Sigma_j^{-1} (x - \mu_j)] \quad (4.4)$$

Grâce aux equations (4.2) et (4.3), il est alors possible de calculer la valeur des Gaussiennes de chaque activité.

Finalement, une activité  $A$  est représentée grâce à cet algorithme par  $A = (\Sigma, \mu, \omega)$ . Ce modèle doit permettre de reconnaître une activité réalisée dans l'appartement.

Les étapes de l'apprentissage d'une activité  $A$  sont:

1. création des vecteurs d'activités  $x$  à partir des jeux de données. Ils contiennent le nombre de fois que chaque capteur est activé pendant la réalisation d'une activité particulière
2. pour chaque vecteur  $x$ 
  - (a) calcul des paramètres des Gaussiennes représentant  $x$ , soit  $\theta_j$ , à l'aide des equations (4.2) et (4.3)
  - (b) calcul de  $p(x|\theta_j)$  (equation (4.4))
3. réitération de l'étape précédente sur chaque distribution  $\theta_j$  représentant  $x$  pour calculer  $prob(x|\Theta)$  (équation (4.1)), ce qui donne une représentation de l'activité  $A$ .

### Estimation des paramètres et algorithme d'Espérance-Maximisation

Maintenant que nous avons étudié le fonctionnement général de l'algorithme, nous allons entrer plus en détail dans son étude pour comprendre comment se fait l'affinement du calcul des paramètres des mixtures, leur initialisation, la définition des différentes classes *etc.*

Soit  $x_{\{i\}}$  les vecteurs de l'activité  $A$ , qui sont des instances de l'activité à reconnaître. L'étape suivante consiste à estimer les paramètres du modèle de  $A$ .

Étant donné qu'une activité est représentée par un modèle statistique de données, il est nécessaire d'estimer les paramètres de ce modèle, qui sont  $\Theta = (\mu, \Sigma)$ . Cette

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

phase correspond à l'étape 2a présentée en 4.3.3. L'estimation de ces paramètres ne peut pas se faire en une seule fois, car nous n'avons pas la connaissance de la distribution qui a pu générer cette activité. Il faut donc utiliser un processus itératif qui permette d'affiner après chaque calcul de probabilité d'appartenance à une distribution la valeur des paramètres. C'est ainsi que l'algorithme 4.1, appelé algorithme d'Espérance-Maximisation (EM), répond à ce problème [41].

---

**Algorithme 4.1** : Algorithme EM

---

**Input** : les vecteurs activité  $x_{\{i\}}$

**Output** : les paramètres  $\mu$  et  $\Sigma$

**begin**

initialiser aléatoirement les paramètres du modèle

**while**  $\mu$  et  $\Sigma$  ne sont pas constants **do**

**Phase d'espérance** Pour chaque vecteur, calculer la probabilité qu'il appartienne à chaque distribution  $j$ , c'est-à-dire calculer  $prob(distribution\ j|x, \theta)$ , via le calcul de  $p_j(x_i|\theta_j)$  et avec l'aide de l'équation (4.5)

**Phase de maximisation** Étant données les probabilités calculées lors de la phase d'espérance, calculer une nouvelle estimation des paramètres  $\mu$  et  $\theta$  qui maximise la probabilité d'appartenance à une distribution,

**end**

**end**

---

Ainsi, en appliquant cet algorithme, on obtient une estimation des paramètres  $\mu$  et  $\theta$  pour chaque vecteur activité  $x$  d'une activité  $A$ , sans savoir quelle distribution a généré ce vecteur. On peut affecter la distribution la plus probable qui a pu générer ce vecteur  $prob(distribution\ j|x, \theta)$ .

Une fois ce calcul fait, on peut en déduire  $prob(x|\Theta)$ , donné par l'équation (4.1), ce qui nous donne la probabilité de  $x$ , donc l'activité la plus probablement réalisée.

#### Initialisation des paramètres

Il existe deux façons principales d'initialiser les paramètres de la mixtures: aléatoirement ou non. L'initialisation aléatoire des paramètres ne donnant pas de bons résultats, nous choisissons de les initialiser grâce à un algorithme de partitionnement de données, *clustering* en anglais. Ce partitionnement va servir à initialiser les diffé-

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

rents poids  $\omega_j$  associés à chaque distribution. L'initialisation est faite au moyen de l'algorithme des C-Moyennes flou, version améliorée des K-Moyennes [22]. Cependant, pour mieux comprendre le fonctionnement de cet algorithme, nous commencerons par expliquer l'algorithme des K-moyennes avant de nous tourner vers l'explication de C-Moyennes flou.

L'algorithme de partitionnement de données va tâcher de rassembler les données qui *se ressemblent* en différentes classes, ou *clusters* en anglais. Ainsi, à chaque objet (dans notre cas chaque instance d'activité) sera affectée une classe qui lui est proche.

**Algorithme des K-moyennes :** L'algorithme de partitionnement de données le plus utilisé est l'algorithme des K-moyennes donné en 4.2.

---

**Algorithme 4.2 :** Algorithme des K-moyennes

---

**Input** : tableau contenant les vecteurs d'activité; nombre de classes désirés

**Output** : les barycentres (un pour chaque classe)

**begin**

Assigner des valeurs aléatoires aux poids de chaque classe

**while** *les barycentres ne convergent pas ou après un certain nombre d'itérations* **do**

- on affecte à chaque objet un barycentre - on met à jour les barycentres en calculant la moyenne des objets qui leur sont associés

**end**

**end**

---

Pour savoir à quoi correspond une classe, nous utilisons la notion de barycentre. Le barycentre est la moyenne des valeurs des objets appartenant à la classe. Si par exemple la classe 1 contient l'objet  $a = \{0, 1, 0, 4, 0\}$  ainsi que l'objet  $b = \{4, 1, 2, 6, 0\}$  alors le barycentre associé à cette classe sera l'objet  $c = \{2, 1, 1, 5, 0\}$

À la fin de l'algorithme, nous obtenons différentes classes regroupant plusieurs objets « proches ». On retrouve au centre de chacune de ces classes un barycentre.

**Algorithme des C-moyennes flou :** Alors que l'algorithme des K-moyennes affecte un barycentre à chaque objet, l'algorithme des C-moyennes flou donné en 4.3 affecte un poids reflétant l'appartenance de cet objet à la classe. Ainsi, dans le cas

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

de trois classes, trois probabilités vont être assignées à chaque objet. Les deux algorithmes fonctionnent de la même façon, mais C-moyennes flou offre une estimation plus précise des poids  $\omega_j$ . En effet, alors que K-moyennes n'offre qu'une estimation binaire des poids (une activité appartient ou non à une classe), C-moyennes flou donne des valeurs d'appartenance allant de 0 à 1 pour chaque classe, ce qui est nettement plus précis. Cette précision est essentielle pour l'initialisation de l'algorithme EM.

---

**Algorithme 4.3** : Algorithme des C-moyennes flou

---

**Input** : tableau contenant les vecteurs d'activité; nombre de classes désiré

**Output** : poids calculés pour chaque classe

**begin**

Assigner des valeurs aléatoires aux poids de chaque classe

**while** *les barycentres ne convergent pas ou après un certain nombre d'itérations* **do**

- calculer les barycentres de chaque classe avec les nouveaux poids

- recalculer les pseudo-partitions des classes c'est-à-dire les poids affectés à chaque classe en minimisant la distance euclidienne

**end**

**end**

---

Un des défauts des méthodes de partitionnement de données, que ce soit K-Moyennes ou C-Moyennes flou, est la nécessité de déterminer le nombre de classes optimal afin que l'activité soit le mieux modélisée possible. Le principe d'associer un nombre plus grand de classes à une activité plus complexe donne une estimation du nombre de classes nécessaires pour chaque activité. Par-exemple la simple activité de se réveiller ne nécessite qu'une seule classe, alors que l'activité de préparer le déjeuner en requiert trois, comme il sera vu plus tard. Dans ce cas, les mixtures de Gaussiennes ne sont justifiées que pour les activités complexes.

Pour connaître le nombre optimal de classes, il suffit d'évaluer à partir des données de validation le taux de reconnaissance pour chaque activité avec différents nombres de classes. Le nombre de classes dépend fortement du jeu de données à partir duquel les activités ont été apprises. À chaque fois qu'une activité est modifiée ou apprise différemment, il faut faire une nouvelle phase d'apprentissage et peut-être changer le nombre de classes nécessaires.

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

Le tableau 4.2 permet de se rendre compte de la corrélation entre le nombre de classes et la complexité d'une activité, obtenu par tests successifs sur les données de validation.

Activité	Nombre de classes
Se réveiller	1
Aller aux toilettes	3
Préparer le déjeuner	3
Manger	1
Faire la vaisselle	2

tableau 4.2 – Nombre de classes en fonction des activités

#### Fonctionnement global de l'apprentissage

La figure 4.3 résume le fonctionnement global du programme d'apprentissage des activités. Il étudie le cas d'une activité particulière  $A$  représentée par une instance  $x$  qui permet de faire l'apprentissage, avec trois classes.

#### 4.3.4 Validation

##### Fonctionnement de la reconnaissance

Une fois que l'apprentissage des activités est terminé, nous connaissons les paramètres  $\omega_j$  et  $p(x|\theta_j)$  pour chaque distribution et pour chaque activité. En appliquant la règle de Bayes, on obtient directement l'équation (4.5), qui donne la probabilité qu'une activité  $A$  représentée par une instance  $x$  vienne d'une distribution particulière  $D_j$ .

$$prob(D_j|x, \Theta) = \frac{\omega_j \cdot p(x|\theta_j)}{\sum_j^K \omega_j \cdot p(x|\theta_j)} \quad (4.5)$$

Ainsi, en reprenant l'équation (4.1), nous pouvons calculer la probabilité qu'une instance d'activité  $x$  appartienne à une mixture. Il suffit d'appliquer cette équation sur toutes les mixtures recueillies lors de l'apprentissage des activités, puis de les comparer

### 4.3. ALGORITHME DE MIXTURE DE GAUSSIENNES

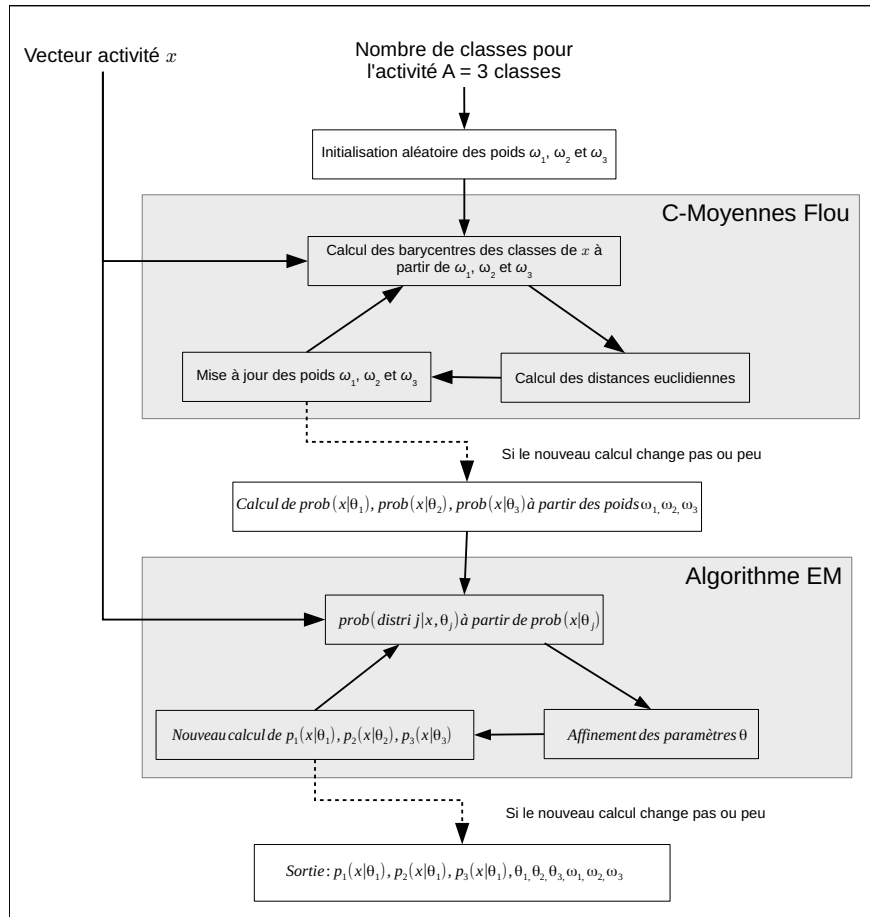


figure 4.3 – Fonctionnement de l’algorithme des mixtures de Gaussiennes avec initialisation des paramètres par C-moyennes flou pour trois classes

entre elles. La probabilité la plus élevée correspond à l’activité la plus probablement réalisée.

Les étapes de la phase de reconnaissance sont donc:

1. création du vecteur de l’activité à reconnaître  $x$  à partir des évènements générés dans l’appartement,
2. Pour l’activité  $A$  apprise lors de la phase d’apprentissage
  - (a) calcul de la probabilité  $prob(distribution\ j|x, \theta)$  (équation (4.5)) qu’une

#### 4.4. BILAN

distribution  $j$  de  $A$  ait été générée à partir de cette activité et des probabilités calculées pendant la phase d'apprentissage,

- (b) calcul de la probabilité qu'une mixture ait pu générer l'activité  $A$   $prob(x|\Theta)$  (équation (4.1)),
3. répétition de l'étape 2 pour toutes les activités apprises, ce qui donne autant de probabilités que d'activités apprises,
4. récupération de la probabilité maximale, ce qui donne l'activité la plus probablement réalisée.

#### Résultats

Lorsqu'on applique l'algorithme de mixture de Gaussiennes avec le nombre de classes indiqué dans le tableau 4.2, sur le jeu de données présenté en section 4.1.2, on obtient les résultats présentés dans le tableau 4.3.

Activité	Nombre total d'instances	Nombre d'instances reconnues	Pourcentage de reconnaissance
Se réveiller	47	47	100
Aller aux toilettes	66	59	89
Préparer le déjeuner	65	39	60
Manger	62	40	64
Faire la vaisselle	67	60	89
Total	307	245	79

tableau 4.3 – Résultats de validation de l'algorithme de mixture de Gaussiennes

Les résultats sont globalement bons, puisque 79% des activités sont reconnues correctement.

## 4.4 Bilan

**Comparaison des résultats :** La figure 4.4 présente les résultats de reconnaissance obtenus pour les deux algorithmes. L'algorithme des mixtures de Gaussiennes répond au problème de la reconnaissance d'activités complexes comme préparer le déjeuner,



#### 4.4. BILAN

puisque la reconnaissance de cette activité est environ quatre fois plus précise qu'avec l'algorithme de simple association.

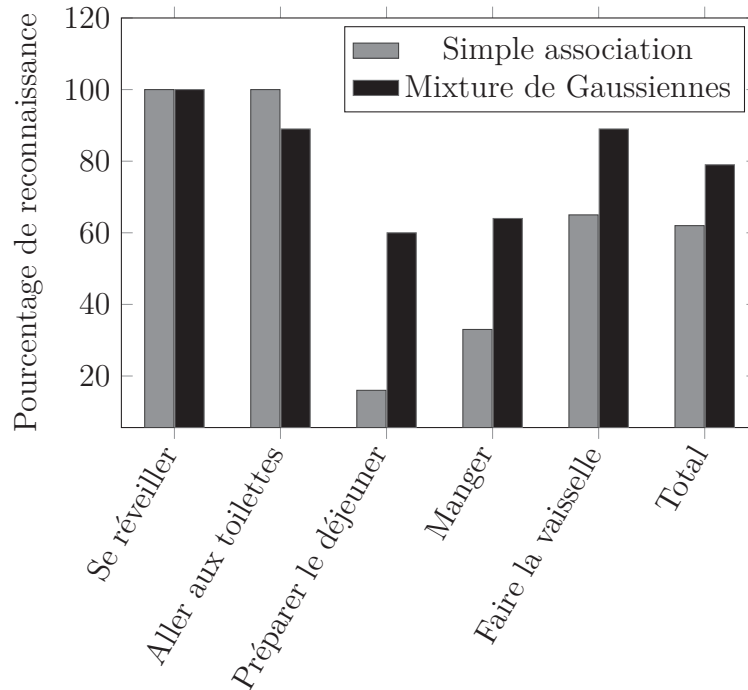


figure 4.4 – Résultats de reconnaissance des deux algorithmes

Pour les activités simples telles que se réveiller, le taux de reconnaissance est bon dans les deux cas. La complexité de l'algorithme des mixtures de Gaussiennes n'est pas justifiée dans ce cas; cependant cette complexité n'empêche pas l'algorithme de les reconnaître correctement.

L'activité d'aller aux toilettes est moins bien reconnue avec l'algorithme de mixture de Gaussiennes qu'avec l'algorithme de simple association. Aller aux toilettes est une activité complexe car elle contient plusieurs scénarios alternatifs: si la personne prend beaucoup de temps, si elle se lave et se sèche les mains après, si elle referme la lunette *etc.* Cependant, cette activité avantage l'algorithme de simple association dans la mesure où c'est la seule activité qui se réalise dans la salle de bain uniquement. Il suffit donc qu'un seul capteur de la salle de bain soit activé pour que l'algorithme l'identifie. L'algorithme de mixture de Gaussiennes va effectuer des calculs plus complexes qui

#### 4.4. BILAN

vont le conduire à faire des erreurs.

L'algorithme de simple association est donc plus approprié pour reconnaître une activité qui se passe dans une pièce unique et dans laquelle aucune autre activité n'est possible. Cependant, il est possible de réaliser d'autres activités dans la salle de bain, comme nous le verrons par la suite, ce qui diminuera le taux de reconnaissance par simple association.

**Contrainte sur le nombre de classes :** Il est possible à l'aide de l'algorithme de mixture de Gaussiennes de reconnaître des activités complexes réalisées dans l'appartement, comparé à l'algorithme de simple association. Cependant, cet algorithme ajoute un certain nombre de contraintes telles qu'une complexité accrue et des paramètres à définir. Il faut notamment définir le nombre de classes nécessaires pour représenter chaque activité. Jusque là, nous déterminions le nombre optimal de classes en faisant des tests successifs avec différentes valeurs. Cependant, plusieurs techniques permettent d'obtenir automatiquement le nombre de classes optimal [50, 33]: les fonctions de Bezdek, de Xie-Beni et de Fukuyama-Sugeno.

Une amélioration à apporter à l'algorithme serait donc d'y inclure une de ces techniques pour ne pas avoir à déterminer soi-même le nombre de classes optimal.

**Fenêtre de reconnaissance :** Une fois l'apprentissage terminé, à chaque activité est associé un modèle correspondant à une combinaison de capteurs représentant l'activité. Cette combinaison a été établie sur une durée fixe des activités à reconnaître, appelée la fenêtre de reconnaissance. Lors de la phase de reconnaissance, l'algorithme de mixture de Gaussiennes requiert de connaître exactement la fenêtre temporelle à analyser pour reconnaître l'activité, contrairement à l'algorithme de simple association qui peut savoir qu'une activité a été réalisée dès qu'un certain nombre d'évènements ont été générés dans l'appartement. Ceci peut induire des erreurs dans la reconnaissance si la fenêtre est mal placée, en étant par exemple à cheval entre deux activités.

En outre, [26] et [21] s'intéressent à la problématique des fenêtres glissantes (section 1.3.2). En attribuant une pondération sur les capteurs, ils créent des fenêtres dynamiques qui leur permettent de reconnaître des activités entrelacées ou concu-

#### 4.4. BILAN

rentes.

La section 5.2.2 examine en détail ce point et en donne une solution.

**Complexité des algorithmes :** La complexité de l'algorithme de mixture de Gaussiennes dépend du nombre d'itérations nécessaires pour obtenir la convergence des paramètres. Deux étapes de l'algorithme contiennent des opérations itératives: l'initialisation des paramètres avec l'algorithme de C-Moyennes flou et l'affinement des paramètres des Gaussiennes avec l'algorithme EM, dont on ne connaît pas de critère de fin. Cette complexité dépend aussi du nombre de classes. Il est donc difficile d'estimer une valeur pour cette complexité. Avec le jeu de données présenté en début de section, on obtient des résultats suffisamment bons et rapides pour estimer qu'il fonctionne en quasi temps-réel.

**Avantages/inconvénients des deux algorithmes :** Pour résumer, le tableau 4.4 compare les différents avantages et inconvénients de chacun des deux algorithmes étudiés.

	Avantages	Inconvénients
Simple association	-Simple  -Efficace si l'activité est identifiable par la pièce	-Peu efficace pour activités longues -Peut être facilement biaisé -Peu efficace pour les activités complexes
Mixture de Gaussiennes	-Reconnaît les activités complexes  -Accorde de la souplesse dans l'apprentissage par le choix de paramètres	-Complexe à mettre en œuvre -Fenêtre de reconnaissance difficile à placer -Nécessite de connaître le nombre de classes

tableau 4.4 – Avantages et inconvénients des deux algorithmes

# Chapitre 5

## Intégration d'AMELIS dans l'habitat intelligent

Lors des chapitres précédents nous avons présenté séparément AMELIS, calendrier électronique capable de rappeler les activités planifiées, et PRADHA, programme de reconnaissance d'activité qui reconnaît des activités à partir d'un jeu de données. Or l'objectif de ce travail est de reconnaître si des activités inscrites sur un calendrier sont réalisées. Dans ce chapitre, il est présenté comment on peut d'une part faire fonctionner PRADHA dans l'habitat intelligent de telle manière qu'il puisse reconnaître les activités réalisées lors de leur réalisation. D'autre part, il est présenté comment faire communiquer les deux programmes ensemble pour qu'AMELIS avertisse PRADHA des activités à vérifier et puisse ainsi émettre les rappels appropriés.

### 5.1 Présentation

La figure 5.1 montre les interactions entre l'utilisateur, AMELIS et PRADHA.

Afin de mieux comprendre le fonctionnement général du programme, nous allons nous placer dans une situation imaginaire où Jean, un patient atteint d'Alzheimer, utilise AMELIS pour se rappeler des activités qu'il a à faire le matin. Il se réveille à l'aide de son réveil tous les matins à 8H00. Une fois qu'il est éveillé, il sait qu'il doit aller aux toilettes mais a tendance à l'oublier s'il est distrait par la radio qu'il allume de façon automatique en se levant. Il a donc planifié dans AMELIS d'aller

## 5.1. PRÉSENTATION

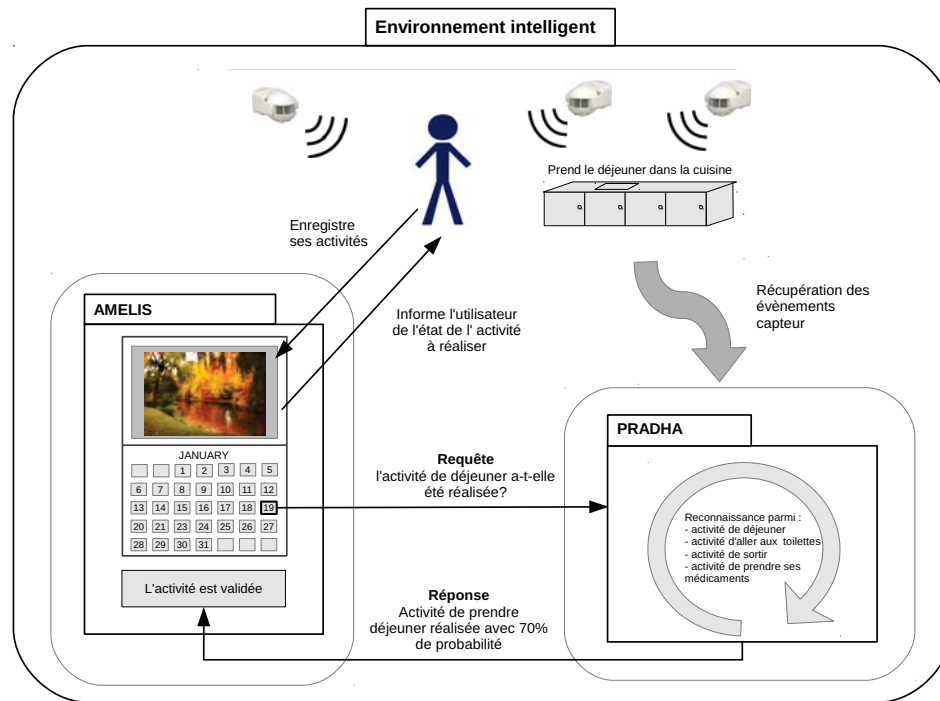


figure 5.1 – Interactions entre AMELIS, PRADHA et l'utilisateur

aux toilettes tous les jours à 8H00. Seulement, certains jours, il se réveille quelques minutes avant son réveil et va aux toilettes directement après s'être levé, soit vers 7H50. Il serait désagréable pour lui de se faire rappeler d'aller aux toilettes à 8H00 alors qu'il vient d'y aller. Pour éviter cela, lorsqu'une activité est prévue à 8H00, AMELIS va être prévoyant et vouloir vérifier ce que fait Jean quelques minutes avant 8H00, soit à 7H45. Cela signifie qu'AMELIS va demander à PRADHA de vérifier si Jean va aux toilettes à 8H00, en commençant à vérifier à partir de 7H45. Si PRADHA reconnaît que Jean n'a pas été aux toilettes avant 8H00, alors il émet un rappel lui rappelant d'aller aux toilettes. Si l'activité n'a pas été réalisée après un certain délais après 8H00, donc à 8H15 par-exemple, c'est parce que Jean a oublié de l'exécuter malgré le rappel. PRADHA émet donc un deuxième rappel à 8H15 pour avertir Jean qu'il a oublié de faire l'activité prévue.

Il est important de noter que c'est PRADHA qui émet les rappels sonores et non AMELIS, ceci afin de diminuer le couplage entre les deux programmes. Cependant,

## 5.2. PRADHA DANS L'APPARTEMENT

étant donné que l'utilisateur a connaissance d'AMELIS et non de PRADHA, il est souvent expliqué que c'est bien AMELIS qui émet le rappel, comme décrit sur la figure 5.1.

On définit une activité  $A$  par cinq paramètres:

- $t_{av}$ , l'heure avant le rappel, qui correspond à l'heure du premier lancement de la reconnaissance d'activité
- $t_r$ , l'heure du rappel, qui correspond à l'heure d'inscription sur AMELIS
- $t_{ap}$ , l'heure après le rappel, qui correspond à l'heure du deuxième rappel si l'activité n'est toujours pas réalisée
- $d$ , la durée moyenne estimée de l'activité.

## 5.2 PRADHA dans l'appartement

L'intégration de PRADHA dans l'appartement se découpe en plusieurs étapes. Tout d'abord, il faut être capable de faire communiquer AMELIS et PRADHA ensemble pour que PRADHA connaisse les activités à rappeler. Une fois que PRADHA sait quelle activité il est sensé reconnaître, qu'il connaît les trois temps  $t_{av}$ ,  $t_r$  et  $t_{ap}$ , il doit récupérer les événements des capteurs générés dans l'appartement pour pouvoir reconnaître cette activité.

### 5.2.1 Récupération des événements des capteurs

Comme vu en section 2.2, grâce à l'utilisation de Maven, il est facile de récupérer les dépendances permettant de se connecter au serveur du laboratoire et ainsi d'être notifié quand un événement se produit dans l'appartement. C'est donc la classe *Demeter* du package *sensorhandler* qui se charge de récupérer les événements. Grâce à ça, à chaque fois qu'un événement est généré dans l'appartement, la classe *Demeter* est notifiée et peut ainsi traiter les données relatives au capteur qui aura été activé.

### 5.2.2 Fonctionnement de la reconnaissance

Dans le chapitre 4, un algorithme de reconnaissance d'activité a été choisi et implanté en Java. Il permet, à partir d'un certain nombre d'événements de capteur générés dans un appartement, de déterminer quelle activité a le plus probablement

## 5.2. PRADHA DANS L'APPARTEMENT

été réalisée. Cependant, cet algorithme ne fonctionne que sur un nombre fixe d'évènements, soit une liste. Cette liste a été générée en fonction de la durée  $d$  de l'activité  $A$ . Finalement, cette liste correspond à une fenêtre de reconnaissance qui a été fixée à un moment dans le temps, au début de l'activité, et dont la largeur correspond à la durée de l'activité.

### Détermination de la plage horaire

Commencer la reconnaissance de l'activité  $A$  suppose de connaître la fenêtre de reconnaissance correspondant à la réalisation de l'activité. Il est possible d'estimer la largeur de la fenêtre grâce à la phase d'apprentissage: pour chaque activité, on établit une moyenne de leur durée de réalisation respective. L'heure du lancement de la première reconnaissance,  $t_{av}$ , est fournie par AMELIS. Cependant, si la fenêtre de reconnaissance est mal positionnée, c'est-à-dire si l'utilisateur ne commence pas l'activité à l'heure prévue, l'activité risque de ne pas être reconnue. Pour éviter un chevauchement entre la fenêtre de reconnaissance et la véritable fenêtre de réalisation de l'activité, il est nécessaire d'obtenir une autre information, qui précise l'heure réelle et non planifiée de début de l'activité. Cette heure de début de réalisation ne peut pas être connue à l'avance puisqu'elle dépend de l'organisation journalière de la personne.

En utilisant les événements clés définis en section 4.2, il est possible de connaître l'heure réelle de réalisation de l'activité. Un événement clé d'une activité  $A$  est défini comme étant un événement qui se produit pratiquement à chaque fois que l'activité  $A$  est réalisée. Si un événement clé est généré, cela signifie que l'activité attendue a été réalisée.

Tel que représenté dans la figure 5.2, trois phases sont nécessaires pour faire la reconnaissance de l'activité  $A$  directement dans l'appartement:

**Phase 1:** AMELIS indique à PRADHA qu'il souhaite savoir si  $A$  est réalisée à partir de la date  $t_{av}$ . Par-contre, rien n'assure que l'utilisateur va effectivement réaliser l'activité à cette heure précise.

**Phase 2:** PRADHA explore les événements de l'appartement à la recherche d'un événement clé de  $A$ , grâce à l'algorithme de simple association.

**Phase 3:** Dès qu'un événement clé de  $A$  est repéré, le programme de reconnaissance

## 5.2. PRADHA DANS L'APPARTEMENT

basé sur les mixtures de Gaussiennes est lancé pendant la durée prédéfinie  $d$  de l'activité.

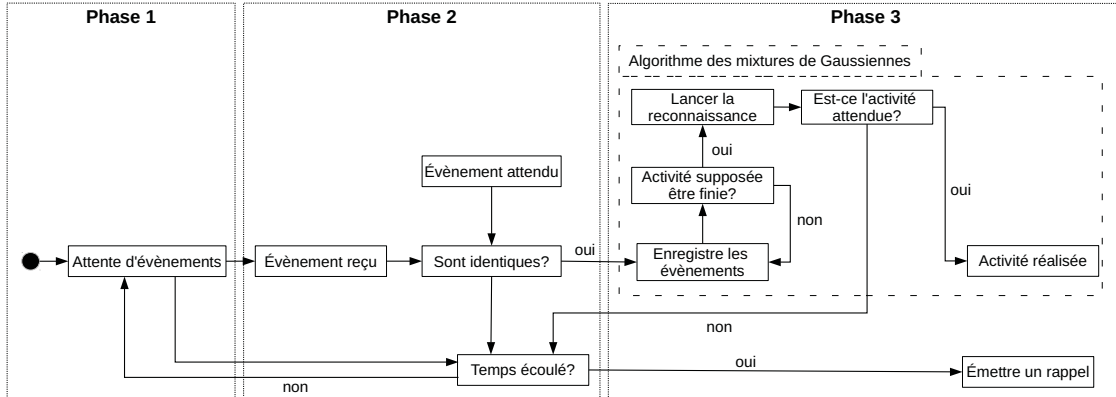


figure 5.2 – Graphe décrivant le fonctionnement de la reconnaissance d'activité dans l'appartement

Le fonctionnement général de la reconnaissance d'activité dans l'appartement revient donc à combiner les deux algorithmes vus en section 4.2 et 4.3: l'algorithme de simple association pour déterminer les évènements clés et donc savoir quand lancer la reconnaissance d'activité proprement dite avec l'algorithme de mixture de Gaussiennes qui sert à vérifier que l'activité a effectivement été faite. Nous pouvons donc ajouter un nouveau paramètre à notre activité:  $L$ , la liste des capteurs clés.

Nous pouvons mettre ce fonctionnement en parallèle avec ce que ferait une personne cherchant une information précise dans un texte: tout d'abord, elle va parcourir vaguement le texte, en cherchant des mots clés correspondant à ce qu'elle cherche. Ces mots clés sont équivalents à nos évènements clés. Une fois un mot clé repéré, la personne va lire le paragraphe contenant ce mot clé avec attention, pour voir si l'information qu'elle cherche s'y trouve. Si oui alors elle arrête la recherche; c'est équivalent à une activité validée. Si non alors elle continue la recherche de mots clés dans le texte, jusqu'à la fin du texte; c'est équivalent à une activité qui n'a pas été réalisée, et pour laquelle on va continuer à attendre sa réalisation, jusqu'à ce que le temps imparti soit écoulé.



## 5.3 Communication entre AMELIS et PRADHA

### 5.3.1 Patron observateur

Afin qu'AMELIS et PRADHA puissent communiquer ensemble sans être couplés l'un à l'autre, un patron de conception très courant en programmation orientée objet est utilisé: le patron observateur.

Lorsqu'en Java une classe est créée, il se peut que la valeur de ses attributs change, et que d'autres classes dépendant de ces attributs soient affectées par ces changements. Dans ce cas il faut que les classes dépendantes soient averties de ce changement. Le patron observateur permet à une classe *observée* d'avertir tous ses observateurs qu'un ou plusieurs de ses attributs a été modifié. Lorsqu'est créé l'objet *observé*, c'est-à-dire l'objet qui va changer d'état, on lui ajoute à sa liste d'observateurs toutes les classes qui seront affectées par ce changement. Ainsi, la classe *observée* n'a pas besoin de connaître les classes qu'elle notifie du changement, et la classe *observatrice* n'a pas besoin de connaître la classe qu'elle observe, seulement les attributs qui sont à même d'être modifiés. Il y n'a donc pas ou peu de couplage entre les deux classes.

Ce patron de conception est utilisé pour faire la connexion entre AMELIS et PRADHA. Lorsqu'il y a une activité  $A$  à rappeler, au temps  $t_{av}$ , la classe *observée* *ReminderModel* va modifier son champ *ActivityToRemind*. Étant donné que *ReminderModel* est observée, elle va notifier tous ses observateurs, dont la classe *AmelisConnection*, que son champ a été modifié. *AmelisConnection* va à ce moment-là lancer la reconnaissance d'activité.

Pour résumer, les étapes à suivre pour lancer la reconnaissance d'une activité  $A(t_{av}, t_r, t_{ap}, d, L)$  sont:

1. À  $t_{av}$ , *ReminderModel* modifie son champ contenant les activités à rappeler, donc le champ *ActivityToRemind*. Ces deux classes sont contenues dans le projet d'AMELIS
2. La classe *AmelisConnection* du projet PRADHA est notifiée qu'il y a eu un changement sur *ReminderModel* et récupère donc la ou les nouvelles activités à rappeler, *ActivityToRemind*
3. À partir de *ActivityToRemind*, on connaît les temps  $t_{av}$ ,  $t_r$  et  $t_{ap}$ , ce qui permet

### 5.3. COMMUNICATION ENTRE AMELIS ET PRADHA

de lancer la reconnaissance aux moments opportuns, ainsi que d'émettre les rappels si nécessaire.

De la même façon que dans AMELIS, le projet *launcher* sert à lancer PRADHA et établit les liens entre tous les autres projets. Il permet notamment de faire le lien entre PRADHA et AMELIS.

#### 5.3.2 Gestion des rappels dans PRADHA

Pour une activité  $A(t_{av}, t_r, t_{ap}, d, L)$ , à partir du temps  $t_{av}$ , PRADHA essaye de reconnaître l'activité  $A$ . Toutefois, si le temps s'écoule jusqu'à  $t_r$  sans que cette activité ne soit réalisée, PRADHA exécute deux actions:

- il relance la reconnaissance de l'activité
- il émet un rappel sonore.

De nouveau, si à  $t_{ap}$  l'activité n'est pas réalisée, PRADHA recommencera ces deux actions.

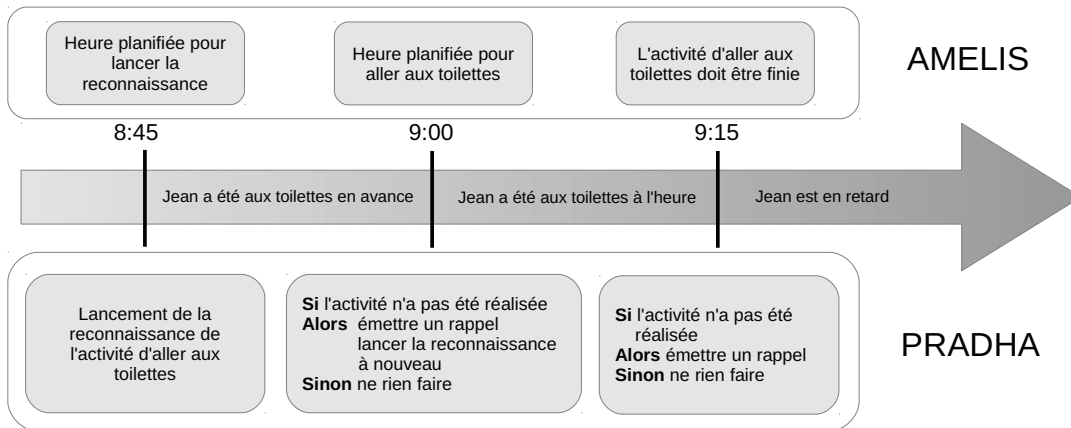


figure 5.3 – Fonctionnement du lancement de la reconnaissance de l'activité  $A$ (« aller aux toilettes »,  $t_{av} = 8:45$ ,  $t_r = 9:00$ ,  $t_{ap} = 9:15$ ,  $d = 3\text{min}$ ,  $L$ )

C'est donc PRADHA qui gère lui-même le lancement de la reconnaissance, en fonction de l'heure donnée par AMELIS.

Pour comprendre le fonctionnement de la gestion des rappels, il faut tout d'abord se pencher sur le diagramme de classes 5.4 qui montre les classes mises en jeu pour

## 5.4. RÉPONSE D'AMELIS À L'UTILISATEUR

gérer les rappels avec leurs interactions.

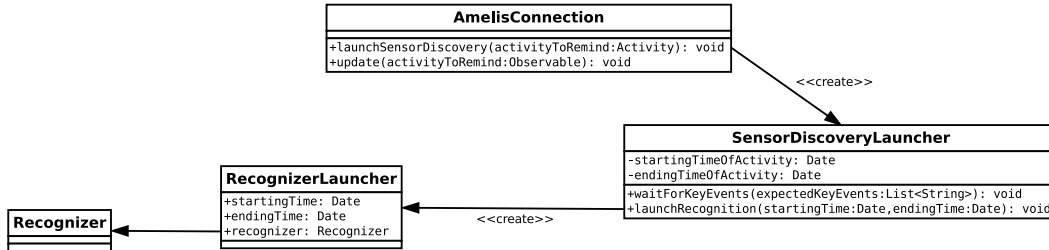


figure 5.4 – Classes permettant de faire de la reconnaissance d'activité

À chaque fois qu'une reconnaissance est lancée, *AmelisConnection* crée une nouvelle instance de *SensorDiscoveryLauncher* qui va permettre de récupérer les événements des capteurs et attendre qu'un des capteurs clé soit activé, connus grâce à la liste  $L$ . Une fois qu'un capteur clé est repéré par le système, *SensorDiscoveryLauncher* crée une instance de *RecognizerLauncher* qui permet d'enregistrer tous les événements et à la fin de la durée  $d$ , lancer la reconnaissance par mixture de Gaussiennes. Si le capteur clé n'est jamais déclenché dans l'appartement, alors la reconnaissance par mixture de Gaussiennes n'est jamais lancée et le programme considère que l'activité n'a pas été réalisée. Finalement, on n'enregistre ni ne lance la reconnaissance par mixture qu'une fois un événement clé déclenché; autrement ces données sont perdues.

L'algorithme permettant de gérer le lancement de la reconnaissance est l'algorithme 5.1.

Avec cette approche, on limite le nombre d'enregistrements et d'analyses des activités: un seul enregistrement est fait une fois qu'un événement clé est détecté, si bien que si l'activité n'est pas réalisée pendant cet enregistrement, la recherche d'événements clés est relancée tant que le temps imparti n'est pas écoulé, ce qui donne une nouvelle tentative à l'utilisateur de faire l'activité attendue.

## 5.4 Réponse d'AMELIS à l'utilisateur

Selon la figure 5.1, AMELIS avertit l'utilisateur de l'activité à réaliser et la lui rappelle s'il a oublié de la faire. Pour cela, plusieurs options sont possibles:

- l'activité à réaliser est affichée à l'écran

#### 5.4. RÉPONSE D'AMELIS À L'UTILISATEUR

---

**Algorithme 5.1** : Reconnaissance des activités dans l'appartement

---

**Require:** Activité attendue  $A(t_{av}, t_r, t_{ap}, d, L)$ , l'évènement  $evt$  généré dans l'appartement, le temps courant  $t$

**while**  $t_{av} < t < t_{ap}$  **do**  
  attend un nouvel évènement  
  **if**  $evt \in L$  **then**  
    **while**  $d$  n'est pas écoulé **do**  
      enregistre les évènements capteur  
    **end while**  
     $A_i \leftarrow$  résultat de la reconnaissance faite avec les évènements enregistrés  
    **if**  $A_i = A$  **then**  
      activité réalisée  
      exit  
    **end if**  
  **end if**  
**end while**

---

- l'activité à réaliser est dictée par un haut-parleur
- les deux à la fois.

Lorsqu'une activité est uniquement affichée à l'écran, l'utilisateur risquerait de ne pas la voir. C'est l'option qui a été choisie pour mettre au courant l'utilisateur à l'avance qu'une activité est à réaliser dans quelques minutes. Cela permet de ne pas être insistant et de rappeler l'activité seulement si l'utilisateur passe devant le calendrier et n'est pas occupé à faire une autre activité. Une copie d'écran de l'interface de l'alarme est donnée en figure 5.5.

Lorsqu'il est l'heure de réaliser une activité et qu'elle n'a pas été faite auparavant, un message vocal rappelle l'activité à faire: « Attention, il est maintenant l'heure de préparer le déjeuner ». Lorsque la personne est en retard, c'est également un message vocal qui rappelle l'activité à faire. Le deuxième message vocal est délibérément insistant, afin de pousser la personne à faire l'activité attendue: « Attention, vous avez oublié de préparer le déjeuner ». On ne souhaite toutefois pas indisposer la personne donc nous avons estimé à deux le nombre de rappels à émettre par AMELIS.

## 5.5. BILAN



figure 5.5 – Interface d’AMELIS pour effectuer le rappel

## 5.5 Bilan

Avec le duo AMELIS-PRADHA, on est capable dorénavant de reconnaître dans l’appartement une activité en train d’être réalisée, en quasi temps-réel. L’utilisation du patron de conception Observateur permet de découpler AMELIS de PRADHA, et autorise ainsi AMELIS à fonctionner sans reconnaissance d’activité. Ceci est essentiel puisqu’il faut permettre à une personne d’utiliser AMELIS sans que son appartement ne soit équipé de capteurs et autres appareils nécessaires à la reconnaissance d’activité.

C’est grâce à la programmation par modules, réalisable grâce notamment à Maven, que l’implantation du programme a été possible: elle a permis de récupérer facilement les événements des capteurs de l’appartement et de faire fonctionner AMELIS conjointement avec PRADHA. Le schéma représentant l’arbre des dépendances permettant à AMELIS et PRADHA de fonctionner ensemble est donné en figure 5.6.

L’arbre inclut les deux algorithmes vus au chapitre 4: simple association et mixture de Gaussiennes. Il inclut aussi l’ensemble du projet AMELIS-PRADHA.

## 5.5. BILAN

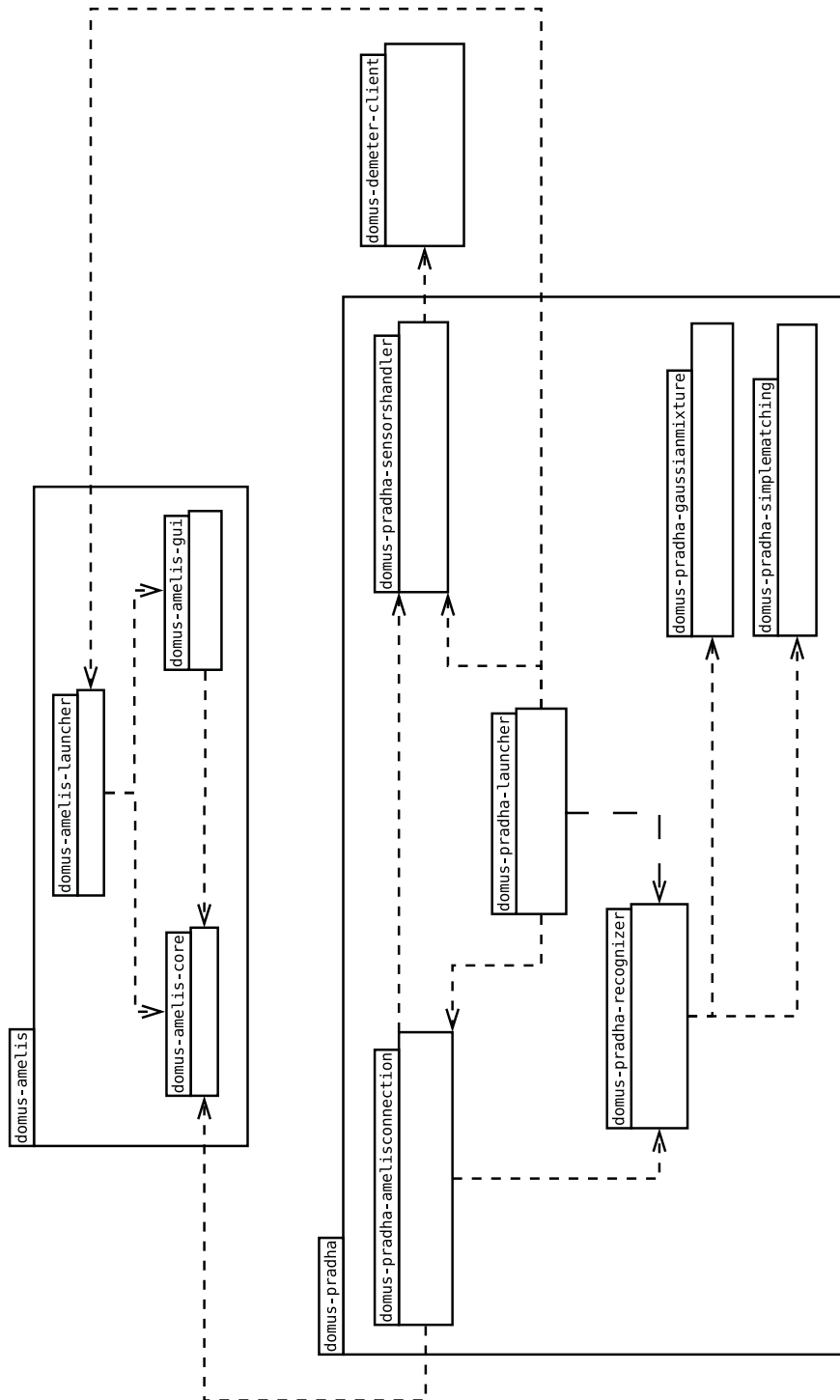


figure 5.6 – Arbre des dépendances d'AMELIS et de PRADHA

# Chapitre 6

## Protocole de l'expérimentation

Pour valider le système AMELIS-PRADHA qui reconnaît les activités faites dans l'appartement telles qu'elles sont inscrites sur le calendrier AMELIS, nous l'avons testé à l'intérieur du laboratoire DOMUS dans des conditions réelles d'expérimentation. Dans ce chapitre, nous allons décrire les buts de l'expérimentation ainsi que son déroulement pour répondre aux objectifs de validation du système.

### 6.1 Buts de l'expérimentation

Comme mentionné en section 2.3.3, le but poursuivi par cette expérimentation est double. D'une part nous cherchons à prouver que l'algorithme de reconnaissance d'activité fonctionne dans des conditions réelles d'expérimentation. Cette évaluation se fait en simplement comptant le nombre d'activités reconnues. Pour cela, un fichier de log est utilisé. D'autre part, nous cherchons à évaluer l'utilisation faite de ce calendrier immergé dans l'appartement, ainsi que la satisfaction de l'utilisateur: comment intègre-t-il le calendrier et le système de reconnaissance d'activité dans le planning de sa journée? Accorde-t-il beaucoup d'attention aux activités rappelées par l'environnement ubiquitaire? Est-il très attentif aux activités rappelées uniquement sur l'écran? Comment réagit-il lorsque des activités déjà réalisées lui sont rappelées ou inversement, si des activités non réalisées ne lui sont pas rappelées? Remarque-t-il quand une activité n'est pas rappelée parce qu'il l'a déjà réalisée?

Étant donné que les buts de l'expérimentation portent sur l'utilisabilité du calen-

## 6.2. EXPÉRIMENTATION

drier en général, il n’y a pas de population particulière ciblée. En effet, bien que le calendrier ait été conçu pour des personnes âgées atteintes de démence, on souhaite tout d’abord évaluer l’utilisation de ce calendrier par des personnes saines. L’expérimentation est finalement une preuve de concept concernant l’utilisation d’un tel système. Les informations recueillies par cette expérimentation servent de première étape pour connaître le potentiel du calendrier immergé dans l’appartement. L’évaluation de l’utilisation par des personnes âgées pourra venir en second étape, en fonction des résultats obtenus à la première.

## 6.2 Expérimentation

### 6.2.1 Population

Douze personnes ont participé à l’expérimentation: quatre femmes et huit hommes, âgés entre 20 et 39 ans. Tous les participants sont des étudiants. Cinq participants vivent seuls, quatre sont en collocation et trois vivent avec leur famille (parents ou conjoint(e)).

### 6.2.2 Déroulement

Le scénario de l’expérimentation est basé sur les activités que l’on réalise habituellement le matin, au réveil. Les activités planifiées dans AMELIS prennent place dans toutes les pièces de l’appartement: trois dans la cuisine, deux dans la salle de bain, une dans le salon et une dans la chambre.

Comme vu au chapitre 5, une activité  $A$  est caractérisée par trois temps, une durée et une liste de capteurs clé:  $(t_{av}, t_r, t_{ap}, d, L)$ . L’heure d’exécution  $t_r$  est l’heure définie dans le scénario et est affichée sur AMELIS. Les deux autres temps  $t_{av}$  et  $t_{ap}$  sont affectés en fonction de l’activité au moment d’ajouter l’activité à l’agenda, à 5-10 minutes avant et après  $t_r$ . Les temps  $t_{av}$  et  $t_{ap}$  sont décidés en fonction de l’intervalle de temps pour lequel on souhaite avoir un rappel. Le déroulement de l’expérimentation est décrit dans le tableau 6.1.

Le participant doit réaliser toutes les activités planifiées sur AMELIS. Celles-ci ont été choisies d’une part en fonction du jeu de données de base tel que discuté en section



## 6.2. EXPÉRIMENTATION

4.1.2 et d'autre part en fonction d'activités supplémentaires apprises à partir d'un jeu de données généré plus tard. Ce jeu de données a permis d'apprendre les activités de prendre ses médicaments et faire le repassage. Ceci donne finalement sept activités reconnaissables par PRADHA. Les autres activités inscrites sur le calendrier AMELIS ont été choisies pour se rapprocher le plus possible de ce que ferait normalement une personne le matin.

L'expérimentation est composée de trois phases:

**Phase 1:** il s'agit d'une phase d'introduction. Elle consiste à présenter l'appartement au participant afin qu'il se repère facilement et puisse exécuter les activités de façon naturelle, comme s'il était chez lui. Pendant cette phase lui sont décrits le déroulement de l'expérimentation et ce qui est attendu de lui. Cependant, le participant ne connaît à ce moment pas encore les activités qu'il doit faire et aura besoin de consulter AMELIS pour le savoir. À la fin de cette phase, on demande au participant d'aller se coucher dans la chambre et d'attendre l'alarme indiquant le début de l'expérimentation.

**Phase 2:** c'est l'expérimentation à proprement dite, qui dure 50 minutes. Elle consiste en l'exécution de 14 activités qui ont été planifiées dans AMELIS. Il est demandé au participant de faire les activités comme s'il était chez lui. L'emploi du temps a spécialement été surchargé pour que le participant soit *débordé* par la quantité de choses qu'il a à faire, le forçant ainsi à manquer des activités, ou à les faire partiellement. AMELIS est alors chargé de lui rappeler les activités manquées. L'emploi du temps surchargé peut également forcer le participant à adapter son agenda en fonction des activités qu'il juge moins importantes, qu'il pourrait abandonner ou faire plus tard. Il ferait donc un de choix des activités à faire en fonction des niveaux de priorité qu'il accorde à chaque activité. PRADHA doit alors reconnaître les activités faites dans des conditions naturelles.

**Phase 3:** il s'agit de la phase de récapitulation. Elle consiste en deux questionnaires:

## 6.2. EXPÉRIMENTATION

Heure planifiée dans AMELIS	Activité	Reconnaissable	Emplacement	$t_{av}$	$t_r$	$t_{ap}$
9:00	Se réveiller		Chambre			
9:03	Aller aux toilettes 1 <sup>re</sup> fois	X	Toilettes	9:00	9:03	9:06
9:04	S'habiller		Chambre			
9:06	Ranger la chambre		Chambre			
9:08	Préparer le déjeuner	X	Cuisine	9:04	9:08	9:12
9:15	Manger le déjeuner	X	Salle à manger	9:12	9:15	9:18
9:20	Nettoyer la table		Salle à manger			
9:21	Atteindre le niveau 7 au jeu Tentrax		Salon			
9:23	Faire la vaisselle	X	Cuisine	9:19	9:23	9:27
9:28	Se brosser les dents		Toilettes			
9:30	Prendre ses médicaments	X	Toilettes	9:27	9:30	9:33
9:32	Se détendre dans le salon		Salon			
9:35	Appeler Pierre		Salon			
9:36	Faire le repassage	X	Chambre	9:33	9:36	9:39
9:45	Aller aux toilettes 2 <sup>e</sup> fois	X	Toilettes	9:43	9:45	9:47
9:50	Aller prendre le bus		Salon			

tableau 6.1 – Programme détaillé de l'expérimentation

## 6.2. EXPÉRIMENTATION

- le premier questionnaire, qui porte sur le profil du participant (âge, style de vie, habitudes dans l’utilisation des agendas). Ce questionnaire a été réalisé pendant la deuxième phase de l’expérimentation où le participant a dû nous appeler par téléphone pour nous donner les informations requises. S’il manquait cette activité, le questionnaire sur le profil de l’utilisateur était à remplir pendant la phase de récapitulation et non pendant l’expérimentation.
- le deuxième questionnaire, qui était à faire remplir par le participant et qui résume son expérience avec le calendrier. Il sert à évaluer l’efficacité, l’utilité et l’utilisabilité du calendrier. Il est notamment demandé si le participant a apprécié l’expérimentation et s’il aimerait avoir un tel système chez lui.

Ces deux questionnaires sont disponibles en annexe B.

Sept activités parmi les 14 planifiées sont reconnaissables par PRADHA. Cela signifie que la moitié des activités planifiées sont des activités non-reconnaissables, c’est-à-dire des activités planifiées dans le calendrier, dont le rappel s’affiche sur AMELIS quelques minutes avant, mais pour lesquelles il n’y a ni rappel vocal, ni reconnaissance d’activité. On n’est donc pas en mesure pour les activités non-reconnaissables de savoir si elles ont été réalisées ou non.

Étant donné qu’un des buts de l’expérimentation est d’évaluer la satisfaction de l’utilisateur, son ressenti par-rapport aux rappels émis quand il est en retard dans la réalisation d’une activité, l’agenda a été volontairement surchargé. En particulier, il a été accordé peu de temps à l’utilisateur pour préparer le repas, le mettant en retard pour l’activité suivante, manger. De même, juste avant l’activité de repassage, il était demandé à l’utilisateur d’appeler au téléphone le laboratoire. L’un de expérimentateurs devait répondre pour le questionner sur ses habitudes, le mettant volontairement en retard pour le repassage. Ceci permet d’évaluer la capacité du calendrier à rappeler des activités en retard.

Même si les activités d’aller aux toilettes sont identiques, elle sont différenciées dans l’agenda et dans l’étude des résultats d’expérimentation car leurs circonstances de réalisation sont différentes: la première au réveil, l’utilisateur dispose de beaucoup de temps, la deuxième à la fin de l’expérimentation, lorsque l’utilisateur doit aller prendre le bus.

## 6.2. EXPÉRIMENTATION

Enfin, l'expérimentation a été dans son intégralité enregistrée par des caméras situées dans la cuisine, le salon et devant AMELIS. Ces caméras permettent d'analyser les activités réalisées, manquées et l'utilisation du calendrier. L'ensemble des résultats est présenté en annexe C.

# Chapitre 7

## Résultats de l'expérimentation

### 7.1 Résultats de la reconnaissance

#### 7.1.1 Données collectées

Les résultats de la reconnaissance portent sur toutes les données qui ont été enregistrées lors de l'expérimentation décrite au chapitre 6. L'ensemble des résultats est présenté en annexe C; nous analyserons dans les sections suivantes la partie de ces résultats jugée la plus pertinente.

Le programme de reconnaissance d'activité PRADHA est lancé à chaque fois qu'une activité inscrite au calendrier AMELIS est supposée être réalisée. Si elle n'a pas été reconnue pendant l'intervalle de temps où elle doit être réalisée, le programme de reconnaissance est lancé une deuxième fois. Pour chaque activité reconnaissable, le programme de reconnaissance est lancé de 1 à 2 fois. Tenant compte des douze participants, cela signifie qu'il y aura au maximum 24 reconnaissances par activité.

Le nombre effectif de reconnaissances lancées par activité pendant l'expérimentation est résumé dans le tableau 7.1.

Grâce au nombre équilibré de reconnaissances d'activités, la précision de l'algorithme peut être validée. La précision de l'algorithme de reconnaissance d'activité est estimée à l'aide de quatre paramètres:

- le nombre de vrais positifs (VP). Un VP correspond à une activité réalisée par l'utilisateur et correctement reconnue par PRADHA. Aucun rappel n'est émis,

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

Activité	Nombre de reconnaissances
Aller aux toilettes - 1 <sup>re</sup> fois	13
Préparer le déjeuner	18
Manger	22
Faire la vaisselle	24
Prendre ses médicaments	19
Faire le repassage	22
Aller aux toilettes - 2 <sup>e</sup> fois	24
Total	142
Moyenne	20.3

tableau 7.1 – Lancements du programme de reconnaissance par activité

à juste titre.

- Le nombre de vrais négatifs (VN). Un VN correspond à une activité qui n’a pas été réalisée et qui est reconnue comme telle. Une alarme est alors émise et rappelle à juste raison que l’utilisateur n’a pas réalisé l’activité attendue.
- Le nombre de faux positifs (FP). Un FP correspond à une activité considérée comme réalisée par PRADHA alors qu’elle ne l’a pas été. Aucun rappel n’est émis alors que l’activité n’a jamais été réalisée.
- Le nombre de faux négatifs (FN). Un FN correspond à une activité non reconnue par PRADHA alors qu’elle a été réalisée. Cela conduit à l’émission d’une fausse alarme puisque AMELIS enverra un rappel pour cette activité alors qu’elle a été réalisée par le participant.

Utilisateur \ PRADHA	Activité reconnue	Activité non-reconnue
Activité réalisée	Activité validée (VP)	Fausse alarme (FN)
Activité non-réalisée	Alarme manquée (FP)	Bonne alarme (VN)

tableau 7.2 – Validation de l’algorithme de PRADHA

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

### 7.1.2 Résultats de l'algorithme complet

#### Précision générale

Tout d'abord, dans le cadre de cette expérimentation, nous évaluons l'algorithme complet comprenant l'algorithme de simple association et de mixture de Gaussiennes fonctionnant ensemble. Ensuite, dans les sections suivantes, les deux algorithmes sont évalués séparément, pour mesurer leur impact sur la reconnaissance totale.

Le tableau 7.3 regroupe les résultats concernant l'algorithme complet, pour tous les participants.

Activité	Vrai positif	Vrai négatif	Faux positif	Faux négatif
Aller aux toilettes - 1 <sup>re</sup> fois	11	1	0	2
Préparer le déjeuner	8	2	0	11
Manger	3	12	0	7
Faire la vaisselle	4	10	2	5
Prendre ses médicaments	4	11	1	6
Faire le repassage	4	15	0	3
Aller aux toilette - 2 <sup>e</sup> fois	9	12	0	2
Total	43	63	3	36
Total (%)	30	43	2	25

tableau 7.3 – Résultats des reconnaissances d'activité

Selon les résultats de reconnaissance obtenus, la précision de la reconnaissance générale ([46]) donne:

$$P = \frac{VP + VN}{VP + VN + FP + FN} = \frac{43 + 63}{43 + 63 + 1 + 38} \simeq 73.1\%$$

Le nombre de VP et de VN dépend du temps accordé au participant pour réaliser cette activité: s'il est pressé par le temps, il est probable qu'il soit en retard et donc qu'AMELIS-PRADHA s'aperçoive qu'il n'a pas réalisé l'activité attendue à temps, engendrant un VN. Notamment, en prenant en compte les VP et les VN, nous obtenons un bon taux de reconnaissance pour l'activité « manger » (17 instances sur 6) car l'algorithme a pu détecter que cette activité était dans de nombreux cas réalisée en

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

retard. Le participant était en effet souvent en retard pour manger puisque le temps accordé par AMELIS pour préparer son repas était court, et l'obligeait à empiéter sur l'activité suivante, « manger » .

On remarque un très faible taux de FP, ce qui signifie que l'algorithme de reconnaissance d'activité n'a presque jamais validé une activité qui n'aurait pas été réalisée. Par-exemple, un FP apparaît dans le cas où le participant était supposé faire la vaisselle mais était encore en train de préparer à manger. Or, pour préparer à manger, il a ouvert le robinet d'eau longtemps; le programme de reconnaissance d'activité a donc cru qu'il faisait la vaisselle, validant ainsi l'activité attendue à tort. Ceci met en évidence l'importance de la connaissance *a priori* de l'heure à laquelle une activité est supposée avoir lieu pour des activités similaires et se déroulant dans la même pièce.

### Précision par activité

En figure 7.1 sont représentées les valeurs des précisions calculées par activité, calculées en pourcentages.

Le plus faible taux de reconnaissance est atteint pour l'activité de préparer le déjeuner. Il s'agit d'une activité complexe où de nombreuses alternatives sont possibles. De plus, certains participants, comme il leur avait été demandé de faire comme chez eux, l'ont exécutée parallèlement à d'autres activités, comme aller aux toilettes ou s'habiller. Par-exemple, ils ont commencé à cuisiner des œufs et du bacon et sont allés aux toilettes pendant la cuisson. Les deux activités ont alors été mélangées et la préparation du déjeuner a été considérée comme non-faite.

L'activité de prendre ses médicaments a un taux légèrement inférieur à 60% car elle a souvent été confondue avec l'activité d'aller aux toilettes, comme elles se déroulent toutes les deux dans la même pièce et que peu de capteurs permettent de véritablement les différencier.

Le détail des résultats est vu dans les sections suivantes où seront étudiés séparément les deux algorithmes.



## 7.1. RÉSULTATS DE LA RECONNAISSANCE

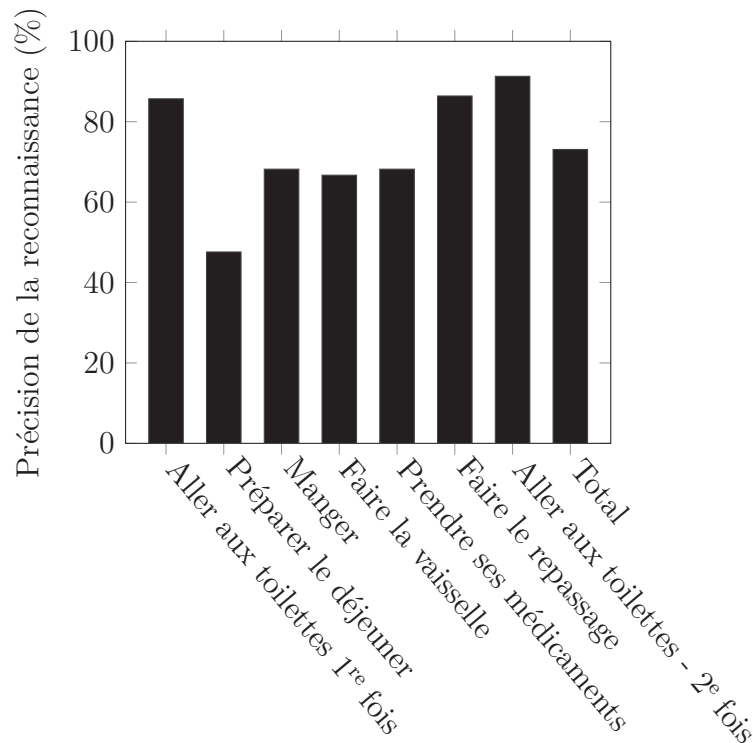


figure 7.1 – Précision de la reconnaissance par activité

### 7.1.3 Résultats de l'algorithme de simple association

Étant donné que l'algorithme de reconnaissance d'activité se divise en deux parties, la partie « découverte » et la partie « identification », une analyse plus fine de la reconnaissance permet de déterminer si chaque algorithme a rempli ses objectifs.

L'algorithme de simple association a pour rôle de détecter l'apparition d'évènements clés. Il ne permet pas de savoir si une activité a été réalisée. Pour cela, il doit être combiné avec l'algorithme de mixture de Gaussiennes. Il est possible de mesurer la précision de cet algorithme (figure 7.2) et ainsi déterminer si cette détection se fait correctement à l'aide des valeurs des VP, VN, FP et FN:

- Un VP correspond à la détection d'un évènement clé appartenant à l'activité planifiée dans le calendrier, indépendamment de si l'utilisateur a réalisé l'activité prévue ou non.
- Un VN est obtenu lorsqu'aucun évènement clé de l'activité attendue n'a été

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

déclenché.

- Un FP est obtenu lorsque le programme détecte le déclenchement d'un évènement clé qui n'est pas en rapport avec l'activité attendue.
- Un FN est obtenu lorsque le programme ne détecte pas d'évènement clé alors que l'activité est en train d'être réalisée.

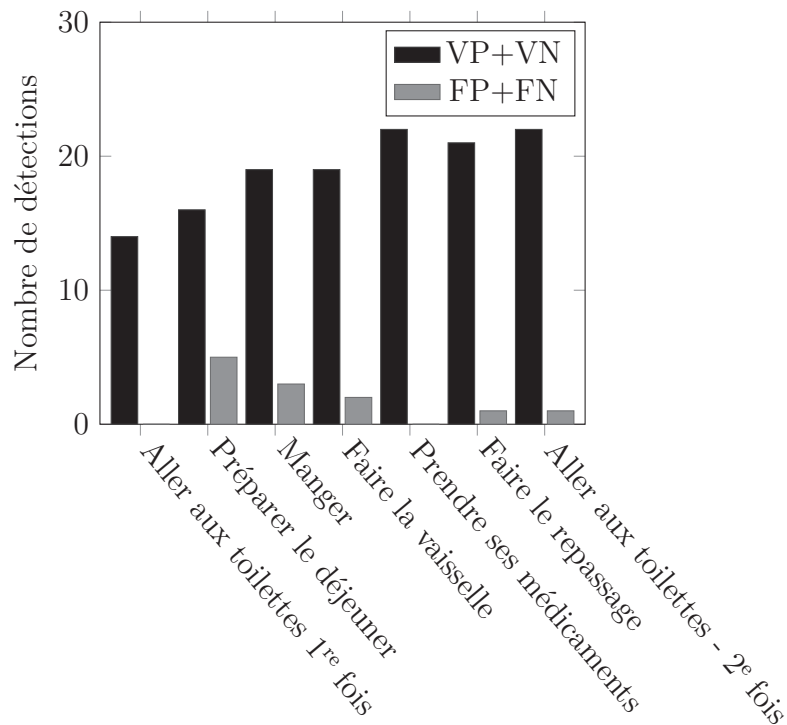


figure 7.2 – Résultats de l'algorithme de simple association

Au total, l'algorithme de simple association obtient une précision de 92%.

On constate que la plupart des évènements clés ont été correctement identifiés. L'activité qui présente le plus haut taux d'erreur est l'activité de préparer le déjeuner. En effet, lors de cette activité, les capteurs mis en jeu influent grandement sur le lancement de la reconnaissance. Il semblerait que certains capteurs aient été mal sélectionnés par l'algorithme d'apprentissage puisqu'ils ne détectent pas toujours le commencement de l'activité. Il se peut de plus que certains capteurs, notamment les capteurs de placards ou de portes de réfrigérateur, ne se déclenchent pas, si la porte a été mal fermée ou pour d'autres raisons inhérentes au capteur. Il est possible de limiter

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

ces erreurs en augmentant le nombre d'évènements clés utilisés par l'algorithme pour détecter le début de l'activité.

De plus, l'activité de préparer le déjeuner est l'activité la plus longue et la plus complexe à réaliser dans l'appartement; de nombreux capteurs sont mis en jeu, mais certains peuvent ne jamais être activés, en fonction de ce que prépare la personne. C'est pourquoi l'algorithme de simple association peut faire des erreurs. Pour améliorer ces résultats et éviter ces erreurs, il faudrait que l'algorithme de simple association choisisse de meilleurs capteurs, plus significatifs, par-exemple en autorisant seulement certains types de capteurs à être des capteurs clé.

Pour ce qui concerne les autres activités, lorsqu'il y a eu des erreurs de détection, c'est généralement dû à des erreurs venant de capteurs non activés, à cause d'une porte cachant le capteur ou un capteur mal orienté. Il peut sinon s'agir de bogues isolés du programme PRADHA dont on n'a pas encore trouvé la source, que l'on pourrait gérer en générant des exceptions.

L'algorithme de simple association remplit correctement son rôle de filtre pour l'algorithme de mixture de Gaussiennes puisqu'il génère peu de FP et FN. Le but est de minimiser le nombre de FN afin de ne pas manquer la réalisation d'une activité. Il serait en effet dommage que l'algorithme de mixture de Gaussiennes ne se déclenche pas à cause d'un capteur clé mal choisi qui ne se déclenche pas au moment voulu. Ce filtre pourrait être amélioré pour l'activité « préparer le déjeuner » en sélectionnant de meilleurs capteurs clé et ainsi éviter les nombreux FN. Il est inutile que le filtre soit trop sélectif puisque dans tous les cas l'algorithme de mixture de Gaussiennes doit s'exécuter pour qu'une activité soit considérée comme réalisée. Il est plus important qu'il puisse détecter une activité potentielle plutôt que de rejeter une activité qui semble ne pas démarrer.

L'utilisation d'un filtre pour détecter le commencement potentiel d'une activité est une solution peu coûteuse et facile à implanter comparée à une solution utilisant une fenêtre glissante. Cette dernière solution lancerait l'algorithme de mixture de Gaussiennes en continu, tout le long de l'intervalle de temps pendant lequel l'activité est supposée se réaliser. Plusieurs problèmes peuvent survenir avec cette solution, comme connaître le nombre de fois que la reconnaissance doit être lancée ou savoir quelle probabilité à accepter au final parmi les nombreuses reconnaissances lancées.

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

Bien qu'imparfaite, la solution du filtre est simple et donne des résultats suffisants dans le cadre des expérimentations menées dans le laboratoire.

### 7.1.4 Résultats de l'algorithme de mixture de Gaussiennes

L'algorithme de mixture de Gaussiennes est lancé une fois que l'algorithme de simple association a détecté le commencement d'une activité. Il a été lancé 94 fois durant l'expérimentation. Nous allons dans cette section étudier son efficacité pour reconnaître une activité (figure 7.3).

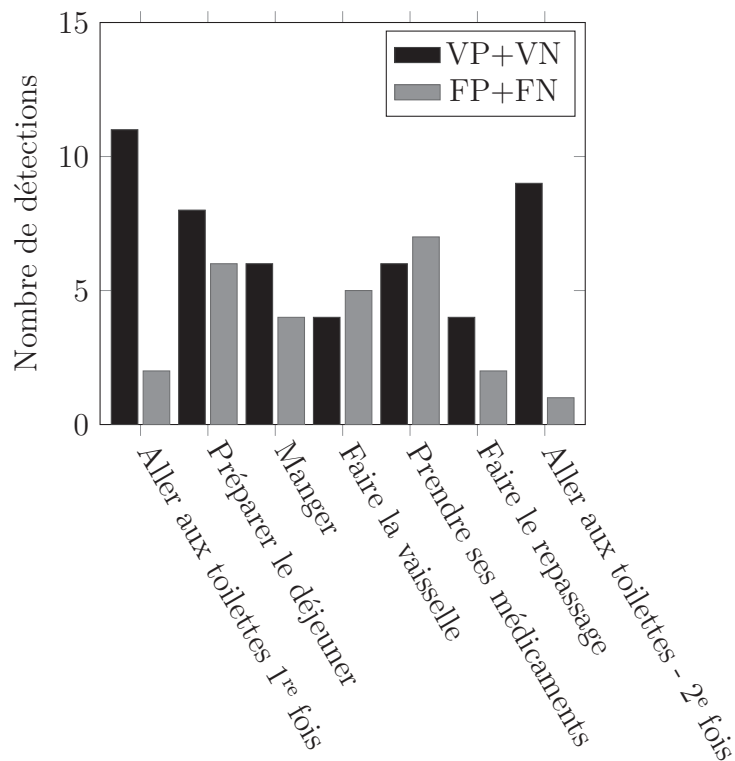


figure 7.3 – Résultats de l'algorithme de mixture de Gaussiennes

Au total, l'algorithme de mixture de Gaussiennes obtient un taux de reconnaissance de 64%, ce qui est un peu plus faible que les résultats obtenus pendant la phase de validation du chapitre 4.3.

## 7.1. RÉSULTATS DE LA RECONNAISSANCE

Bien souvent, les participants faisaient des aller-retours entre la cuisine et la salle à manger, ce qui avait tendance à activer un des capteurs clé de la salle à manger. La reconnaissance de l'activité de manger était alors lancée avec les mixtures de Gaussiennes alors que l'utilisateur n'était pas en train de manger. Comme le montrent les résultats, peu de FP et FN ont été générés pour l'activité « manger », l'algorithme a donc été capable de ne pas la confondre avec « préparer à manger ». L'algorithme a donc rempli son rôle en faisant la différence entre deux activités proches.

Pour ce qui concerne la prise de médicaments, l'algorithme avait tendance à la confondre avec l'activité « aller aux toilettes » si l'utilisateur se rapprochait trop du siège d'aisance. L'algorithme de simple association a rempli son rôle en indiquant que la personne était entrée dans la salle de bain et donc était potentiellement en train de prendre ses médicaments comme prévu. Cependant, l'algorithme de mixture de Gaussiennes avait tendance à infirmer cette proposition en déduisant que l'activité réalisée était celle d'aller aux toilettes plutôt que de prendre ses médicaments. Une façon d'éviter ces erreurs serait de prendre ses médicaments dans une autre pièce ou d'utiliser un tiroir ou un pilulier doté d'un capteur pour ranger les médicaments, qui s'activerait à chaque prise.

Les résultats obtenus pour l'algorithme de mixture de Gaussiennes dépendent du placement de la fenêtre de reconnaissance, donc de l'algorithme de simple association. Étant donné que l'algorithme de simple association est considéré comme peu efficace pour l'activité de préparer le déjeuner, il est très probable que la fenêtre de reconnaissance soit mal positionnée, et donc que le début de cette activité soit mal identifié. Ceci engendre une mauvaise reconnaissance de la part de l'algorithme de mixture de Gaussiennes. D'autre part, la largeur de la fenêtre est déterminée de manière fixe, en fonction des durées moyennes d'activités obtenues lors de la phase d'apprentissage; essayer avec d'autres largeurs de fenêtre permettrait d'améliorer ces résultats.

Concernant les autres activités, mise à part la prise de médicaments, les taux de reconnaissance sont bons; la fenêtre de reconnaissance est donc bien placée et permet une bonne reconnaissance en général. L'algorithme de simple association remplit donc bien son rôle pour positionner les fenêtres de reconnaissance.

## 7.2. RÉSULTATS D'UTILISABILITÉ DU CALENDRIER

### 7.1.5 Bilan

En conclusion, les résultats de la reconnaissance sont relativement bons mais pourraient être améliorés pour certaines activités complexes. Ceci est principalement dû à la liberté accordée aux participants. En effet, même si ceux-ci devaient se conformer à l'horaire, la consigne était de réaliser les activités comme s'ils étaient chez eux. Ceci engendre une certaine complexité, car certains participants ont réalisé des activités en parallèle et pour les activités complexes, le mode de réalisation était propre à chacun. Cependant, un des buts de l'expérimentation était de permettre aux gens de se sentir libres dans le moyen de réaliser les activités, de se sentir chez eux, comme dans la vie réelle; ceci a une incidence sur les résultats de reconnaissance, mais permet d'être réaliste.

Par-ailleurs, l'expérimentation ne poursuit pas comme seul but d'obtenir un fort taux de reconnaissance. Comme mentionné au chapitre 2, elle sert aussi à déterminer comment une personne utilise un calendrier qui lui rappelle les activités importantes, ainsi que l'influence de ce calendrier sur l'organisation de sa matinée. Ceci est étudié en section suivante.

## 7.2 Résultats d'utilisabilité du calendrier

Pour analyser l'utilité du calendrier, deux types d'activités ont été planifiées dans AMELIS: des activités reconnaissables, celles que nous venons d'étudier dans la section précédente, et des activités non-reconnaissables. Les activités non-reconnaissables ne sont pas rappelées vocalement, ce qui va nous permettre d'étudier l'influence d'un rappel sur la réalisation des activités. Un autre paramètre important entre en jeu dans l'exécution des activités planifiées dans l'agenda: leur importance, qui dépend du jugement de chaque utilisateur.

### 7.2.1 Analyse des résultats

**Niveau d'importance des activités** Dans le questionnaire final, il a été demandé à chaque participant d'indiquer pour chaque activité leur niveau d'importance. En faisant les moyennes des niveaux d'importance annoncés, puis en les arrondissant,

## 7.2. RÉSULTATS D'UTILISABILITÉ DU CALENDRIER

nous obtenons les résultats présentés dans le tableau 7.4.

Activité	Niveau d'importance
Se réveiller	3
Prendre le bus	3
Prendre ses médicaments	3
Appeler Pierre	2
S'habiller	1
Ranger la chambre	1
Nettoyer la table de la salle à manger	1
Se brosser les dents	1
Aller aux toilettes	1
Manger	1
Faire la vaisselle	1
Faire le repassage	1
Préparer le déjeuner	0
Jouer au jeu Tantrix	0
Se détendre dans le salon	0

tableau 7.4 – Moyenne arrondie des estimations des niveaux d'importance des activités

Selon les participants, les activités qui ont le niveau d'importance le plus élevé sont les activités qui nécessitent un horaire précis (comme prendre le bus ou se réveiller) ou alors qui ne doivent pas être manquées (comme prendre ses médicaments). Les AVQ n'ont en général pas un niveau d'importance élevé puisque la plupart des personnes estiment qu'elle se réalisent « naturellement » et qu'elles n'ont donc pas besoin de rappel. C'est à ce niveau-là que les résultats diffèreraient d'une personne âgée atteinte d'Alzheimer qui aurait besoin qu'on lui rappelle de se brosser les dents ou de manger.

**Influence du rappel sur la réalisation des activités** Parmi les activités rappelées par un signal sonore, une seule n'a pas été réalisée (figure 7.4). De plus, les activités réalisées en retard sont le plus souvent des activités qui n'ont pas été rappelées avec un signal sonore. Cela signifie que le rappel pousse l'utilisateur à réaliser ses activités, en particulier le deuxième rappel puisqu'il évite à l'utilisateur de manquer une activité. La plupart des activités réalisées à l'heure étaient sans rappel sonore. Ceci est explicable car un autre paramètre entre en jeu dans la réalisation des activités: leur enchaînement donné par l'agenda. Lorsque la personne dispose de temps,

## 7.2. RÉSULTATS D'UTILISABILITÉ DU CALENDRIER

elle peut consulter l'agenda et donc réaliser l'activité prévue, qu'elle lui soit rappelée ou non. Si par-contre elle manque de temps, elle ne prendra pas la peine de consulter son agenda et préférera se fier aux rappels sonores pour connaître l'activité suivante à réaliser. C'est pourquoi pratiquement aucune activité avec rappel sonore n'a été manquée.

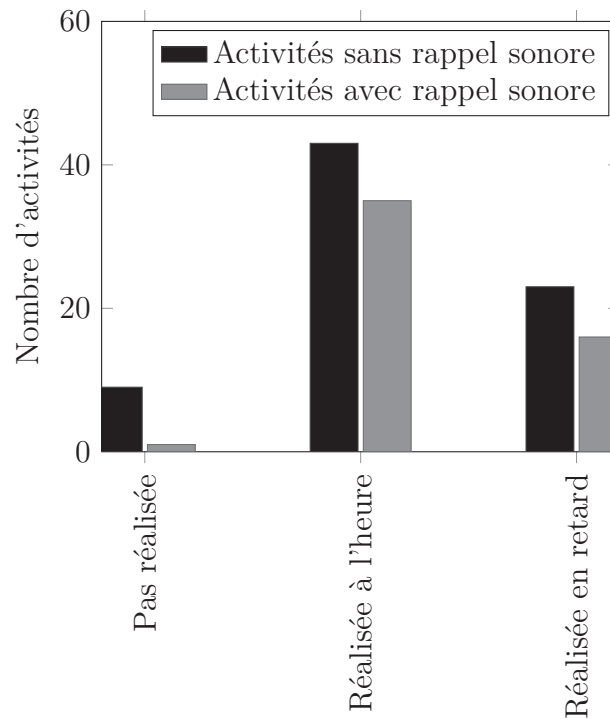


figure 7.4 – Nombre d'identifications de réalisation des activités par PRADHA

**Transparence du calendrier** Il a été demandé à chaque participant d'indiquer s'il avait compris dans quelle mesure certaines activités avaient été rappelées vocalement et d'autres non. La plupart (huit personnes sur douze) répondaient que les activités rappelées vocalement étaient selon eux les activités les plus importantes. Ils étaient alors surpris qu'on ne leur ait pas rappelé l'activité de prendre le bus. Nous leur avons donc expliqué que les activités rappelées étaient les activités que nous étions capables de reconnaître dans l'appartement et donc capables de rappeler uniquement si elles n'avaient pas été réalisées. Les participants ont alors avoué ne pas s'être rendu



## 7.2. RÉSULTATS D'UTILISABILITÉ DU CALENDRIER

compte qu'on ne leur avait pas rappelé des activités telles que préparer à manger, parce qu'ils les avaient faites en avance. Le rappel adapté à ce que fait la personne dans l'appartement évite donc à l'utilisateur d'avoir des rappels inutiles et insistants, sans toutefois surprendre l'utilisateur par son « silence ».

**Confiance accordée au calendrier** Nous nous sommes aperçus que les utilisateurs accordaient une certaine confiance à AMELIS, notamment pour ce qui était de rappeler les activités jugées importantes. Par-exemple, l'activité de sortir et prendre le bus, qui a le plus haut niveau d'importance (tableau 7.4), est une activité qui a été exécutée en retard sept fois sur douze, soit plus d'une fois sur deux. Lorsque nous interrogeons le participant sur son retard, il répondait qu'il pensait qu'AMELIS le lui rappellerait et donc il se permettait de se détendre dans le salon en attendant qu'AMELIS émette un rappel. Mais dans l'expérimentation, comme pour toutes les activités non-reconnaissables, seul un rappel visuel était effectué pour cette activité, d'où le retard.

**Intérêt pour un tel calendrier** La plupart des participants (neuf participants sur douze) ont manifesté un certain intérêt pour un tel calendrier. Ils trouvaient utile l'émission du rappel en fonction de ce qui est fait dans l'appartement, mais pas pour les AVQ (excepté un participant qui le trouvait utile pour tous les rappels). En effet, le système leur apparaissait pratique pour une activité telle que le réveil où AMELIS sonnerait plusieurs fois jusqu'à détecter que la personne s'est levée. Ils le trouvaient également pratique pour l'activité de prendre le bus car il leur permettait de faire autre chose en attendant l'heure du bus, comme regarder la télévision, ce qui leur éviterait de regarder l'heure constamment pour savoir quand partir. Trois utilisateurs ont apprécié qu'on leur rappelle des activités désagréables à faire telles que la vaisselle ou ranger la table; ils ont avoué que, sans le rappel, ils ne l'auraient probablement pas fait. Cependant, pour ce qui est de la majorité des AVQ, ils trouvaient bien souvent les rappels inutiles car ils s'apprêtaient à faire ce que le calendrier leur rappelait. De plus, le fait de rappeler deux activités qui se suivaient de façon logique et naturelle était souvent perçu comme désagréable, comme par exemple l'activité « préparer à manger » suivie de l'activité « manger ».

## 7.2. RÉSULTATS D'UTILISABILITÉ DU CALENDRIER

Par-ailleurs, huit participants sur douze connaissent des personnes pour qui ce genre de système pourrait être utile: grand-parents, personnes déficientes et personnes en manque d'autonomie en général. Elles estimaient que le rappel de la prise de médicaments et des AVQ en général pourraient être bien utiles à ce type de personnes.

### 7.2.2 Bilan

L'expérimentation a été réalisée avec des personnes jeunes et saines. Le rappel oral des activités par le calendrier a un impact sur leur réalisation par rapport à celles uniquement rappelées à l'écran, surtout lorsque la personne consulte peu son agenda. De plus, le rappel d'AVQ a été jugé lourd et peu utile par la plupart des participants. Ils trouvaient cependant le système agréable à utiliser et voyaient son potentiel pour des activités nécessitant de la ponctualité, pour les activités importantes ou encore pour des utilisateurs atteints de déficiences cognitives.

# Conclusion

L'informatique ubiquitaire vise entre autres à assister les personnes dans la réalisation de leurs tâches quotidiennes, en particulier les personnes atteintes de troubles cognitifs. C'est probablement l'avenir de notre habitat d'être de plus en plus automatisés, de plus en plus interactifs avec l'utilisateur et surtout de plus en plus attentifs à ce qui se réalise en son sein. L'habitat intelligent du laboratoire DOMUS s'intègre à cette approche en fournissant un environnement doté de capteurs, d'écrans et d'appareils divers facilitant les tâches de la vie quotidienne.

Le maintien à domicile des personnes âgées est une des solutions privilégiées comparativement aux maisons de soins. Elle requiert cependant de nombreuses ressources humaines et peut être très coûteuse en temps et en argent, ce qui la rend in-envisageable dans certains cas. L'informatique ubiquitaire peut aider à ce maintien, mais pour l'instant peu de projets se sont révélés vraiment réalisables. Ils peuvent être ou bien trop coûteux en matériel, ou bien trop contraignants pour le patient, ou encore non adaptés aux besoins de la personne. De nombreux projets de par le monde tentent de répondre au problème de l'assistance domiciliaire au moyen de la technologie. Aucun n'aboutit à une solution complète, mais chacun apporte sa pierre à l'édifice. C'est également le but de ce projet.

Avec AMELIS-PRADHA, nous avons tenté d'offrir un calendrier électronique intégré à l'habitat intelligent capable de rappeler les activités planifiées en fonction de ce qui est fait dans l'appartement. Ce projet exige de relever plusieurs défis. Tout d'abord, créer un calendrier électronique sur un écran tactile utilisable par une personne âgée, nécessitant peu d'apprentissage de sa part pour pouvoir s'en servir. Ce n'est pas un mince défi car l'équilibre est difficile à gérer entre la facilité d'utilisation et le nombre de fonctionnalités disponibles sur le calendrier. Ensuite, une phase de

## CONCLUSION

reconnaissance d'activité est nécessaire. De nombreux laboratoires se sont penchés sur ce problème et proposent diverses solutions utilisant des caméras, des étiquettes RFID ou différents types de capteurs. Pour éviter toute contraintes sur l'utilisateur, limiter les coûts et respecter la vie privée, seuls les capteurs sont utilisés pour faire de la reconnaissance d'activité dans le cadre de ce projet.

Cette reconnaissance d'activité implique d'établir un modèle pour chaque activité que l'on souhaite reconnaître. Ce modèle doit être à la fois générique, c'est-à-dire un modèle de représentation des activités de haut-niveau, à la fois multi-utilisateurs, c'est-à-dire valable pour n'importe quelle personne réalisant les activités dans l'appartement. Ces défis sont importants et nécessitent une phase d'apprentissage minutieuse, avec des algorithmes complexes qui ont déjà fait leurs preuves dans certains domaines comme la reconnaissance de la parole ou de caractères, mais pas forcément dans la reconnaissance d'activité. Grâce à PRADHA, nous avons toutefois été en mesure d'implanter un programme répondant à ces objectifs et donnant des résultats satisfaisants.

La communication entre AMELIS et PRADHA est donc un avantage pour faire de la reconnaissance d'activité et a bien été exploitée dans le cadre de ce projet; des améliorations sont toutefois possibles afin de mieux utiliser l'information fournie par AMELIS.

Les informations déduites par PRADHA concernant la réalisation des activités dans l'appartement permettent d'émettre un rappel si la personne n'a pas réalisé l'activité attendue, ou de ne rien faire dans le cas contraire. Il a été possible de valider le fonctionnement de ce système de rappel d'AVQ grâce à une expérimentation menée dans le laboratoire avec des personnes jeunes. Cette expérimentation est une première étape pour s'assurer d'une part que l'algorithme de reconnaissance d'activité fonctionne correctement dans un habitat intelligent, dans des conditions de vie réelle, d'autre part pour obtenir un premier avis sur l'intérêt d'un tel calendrier. AMELIS-PRADHA a été apprécié des utilisateurs, qui ont trouvé en ce calendrier surtout utile pour rappeler des activités importantes ou nécessitant de la ponctualité. L'utilité pour le rappel des AVQ leur a paru se limiter aux activités considérées comme pénibles à faire, telles que la vaisselle ou le rangement. Cependant, pratiquement tous les participants à l'expérimentation ont considéré qu'un tel calendrier serait utile pour

## CONCLUSION

des personnes âgées ou atteintes de déficiences cognitives. La prochaine étape serait de faire des expérimentations avec cette clientèle, afin de vérifier l'efficacité et le réalisme d'un tel projet. La combinaison d'un calendrier avec un programme de reconnaissance d'activité est donc une solution prometteuse, du moment qu'on arrive à avoir un programme de reconnaissance d'activité efficace et fiable.

Il a été choisi de faire un programme de reconnaissance d'activité qui soit indépendant de la personne qui utilise le calendrier. Ce choix n'est pas anodin puisqu'il permettrait, pour une configuration d'appartement donnée, de mettre en place ce système dans plusieurs appartements intelligents à la fois, sans avoir à refaire toute la configuration du programme, comprenant la phase d'apprentissage qui est une phase longue et laborieuse. Ceci pourrait notamment être utilisé dans le cadre des appartements alternatifs qui s'apprêtent à être mis en place à Sherbrooke. Il s'agit d'une dizaine d'appartements intelligents, tous équipés de la même façon, avec exactement la même configuration, dans lesquels vivent des personnes atteintes de traumatismes crâniens. Cette infrastructure permettra d'effectuer des expérimentations et des tests grâce aux nombreuses données recueillies. On pourrait alors imaginer qu'AMELIS-PRADHA soit mis en place dans chacun de ces appartements. Cette mise en place serait alors rapide à faire car identique pour chaque appartement, grâce aux choix d'implantation du programme de reconnaissance d'activité.

Pour que AMELIS-PRADHA soit véritablement utilisable et efficace dans l'assistance à la personne, il est important d'augmenter le nombre d'activités reconnaissables dans l'appartement. Jusque-là, ce nombre était limité au jeu de données déjà existant dans le laboratoire, mais en utilisant un nouveau jeu de données annoté, on pourrait apprendre de nouvelles activités. Ceci peut facilement être fait car PRADHA accepte de faire l'apprentissage à partir de n'importe quel fichier texte, dès lors que celui est correctement formaté. La difficulté survient lorsqu'il s'agit de déterminer le nombre de classes nécessaires pour représenter chaque activité. Comme mentionné au chapitre 4, des algorithmes permettent d'obtenir automatiquement le nombre de classes optimal pour représenter une activité. Cependant, aucun n'a été implanté dans le cadre de cette maîtrise; le nombre de classes optimal a été déterminé à l'aide de tests successifs. Ce serait donc une amélioration notable que, lors de la phase d'apprentissage des activités, le nombre de classes optimal soit déterminé automatiquement, en fonction

## CONCLUSION

de l'activité à apprendre.

Contrairement à d'autres projets de reconnaissance d'activité où le programme n'a absolument aucune connaissance de l'activité qu'il est supposé identifier, le projet AMELIS-PRADHA a l'avantage d'avoir une idée, grâce à ce qui est programmé dans AMELIS, de l'activité à analyser. Une activité programmée dans AMELIS contient l'information sur son début, et par conséquent sur sa fin, puisque sa durée est connue grâce à la phase d'apprentissage. Ces deux informations permettent alors d'établir une fenêtre de reconnaissance fixe, facilitant grandement la reconnaissance d'activité en analysant les événements capteurs générés dans cette fenêtre de temps. Cependant, lors des expérimentations menées dans le laboratoire, nous nous sommes aperçu que la taille de cette fenêtre pouvait poser des problèmes, notamment lorsque l'activité supposée être réalisée chevauche deux fenêtres fixes. En fonction de ce chevauchement, la reconnaissance peut alors être erronée. Une amélioration possible serait, comme il a été suggéré dans la revue de littérature au chapitre 1, d'utiliser des fenêtres dynamiques variables qui s'adaptent en fonction des événements générés dans l'appartement et des activités réalisées.

Il devrait être possible de rapprocher le fonctionnement de AMELIS-PRADHA de celui de Autominder [39], en offrant un système de pondération des activités: activité importante, peu importante, facultative, à réaliser à l'heure absolument *etc.* Ceci permettrait d'adapter les rappels automatiques non seulement en fonction de ce qui est à faire dans l'appartement, mais aussi en fonction du poids de l'activité. Ainsi, une activité peu importante ne serait pas rappelée aussi souvent qu'une activité importante. Des algorithmes d'intelligence artificielle seraient à mettre en place pour permettre un rappel utile et réellement adapté à l'environnement de la personne.

Le programme AMELIS-PRADHA a été conçu suivant des patrons de conception connus en Java. Il a été découpé en plusieurs modules, permettant l'ajout, la suppression et le remplacement aisé de certains modules, ceci dans le but de compléter, d'améliorer chacun des deux programmes. Par-exemple, il devrait être facile d'ajouter un module de météo qui irait chercher automatiquement sur Internet la météo et l'afficherait sur l'écran d'AMELIS.

Il pourrait aussi facilement être envisageable d'utiliser le module de reconnaissance d'activité sur un autre jeu de données que celui utilisé dans cette maîtrise, et donc

## CONCLUSION

faire de la reconnaissance d'activité, mais dans un autre cadre qu'AMELIS-PRADHA, en récupérant les parties du code intéressantes. Nous espérons donc avec ces deux projets offrir une base aux prochains étudiant-chercheurs; base qu'ils pourront modifier, utiliser, compléter ou dont ils pourront s'inspirer pour leur propre projet.

Le projet AMELIS-PRADHA, grâce à l'expérimentation, a pu fournir une preuve de concept concernant l'utilité d'un tel système et l'influence qu'il pourrait avoir sur la réalisation des AVQ. En augmentant le nombre d'activités que le système serait capable de reconnaître et en améliorant certains points du programme pour reconnaître plus efficacement certaines activités, il serait tout à fait envisageable qu'il apporte une réelle assistance à la personne. Cette assistance peut non seulement se traduire par des rappels d'activité efficaces, mais aussi par des détections d'anomalies, pouvant entraîner des appels d'assistance automatiques. On espère pouvoir étendre cette preuve de concept à un nouveau calendrier intelligent permettant aux personnes âgées de rester chez elles plus longtemps, sans toutefois nuire à leur vie privée et à leur sécurité.

Les résultats de l'expérimentation indiquent que les participants voyaient en AMELIS-PRADHA, en plus d'un intérêt pour les personnes âgées, un intérêt pour eux-mêmes pour les activités importantes et nécessitant de la ponctualité. On peut alors imaginer étendre un tel système à de nombreux autres domaines d'application où les rappels d'activité sont importants, que ce soit à domicile ou même dans le domaine professionnel.

# Annexe A

## Diagramme de classe d'AMELIS

Ci-dessous sont représentés deux diagrammes de classes décrivant le fonctionnement d'AMELIS pour l'édition et la planification des activités. En particulier ils permettent de comprendre le fonctionnement des interactions avec l'utilisateur à l'aide du modèle MVC.



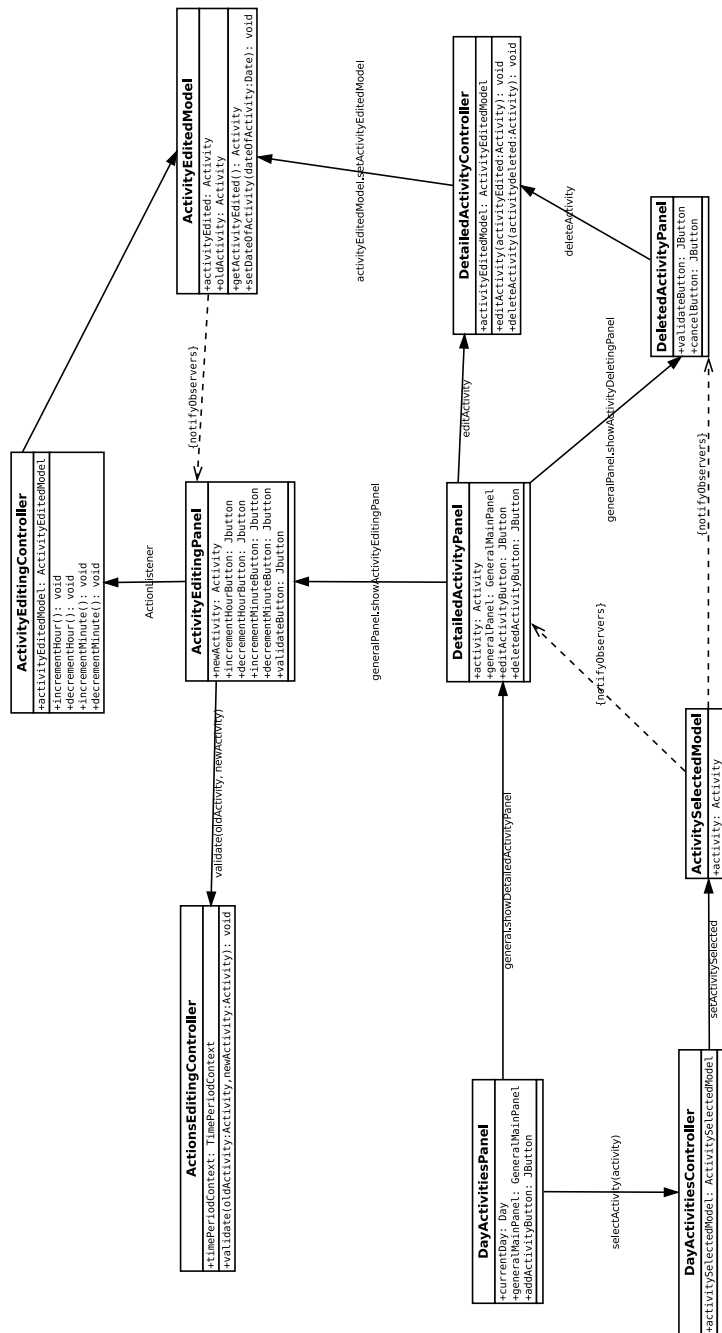


figure A.1 – Diagramme UML représentant l'édition d'une activité

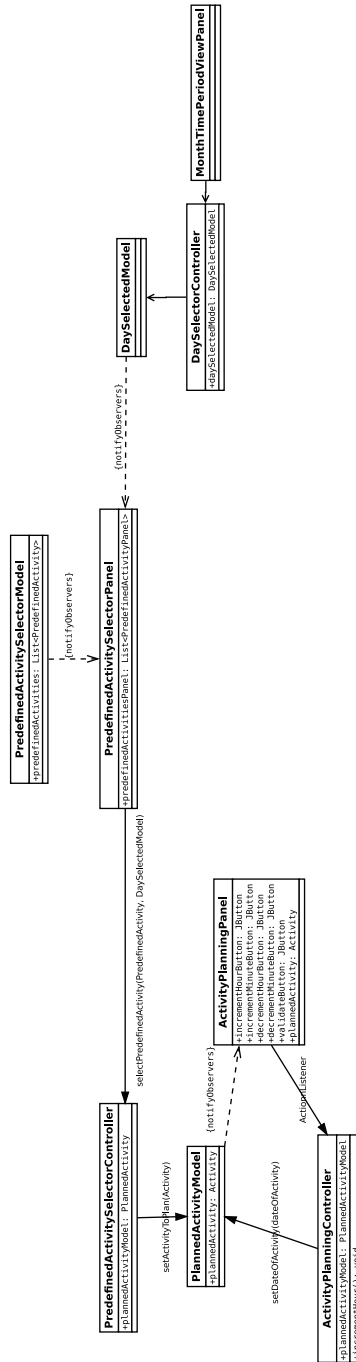


figure A.2 – Diagramme UML représentant la planification d’une activité

# Annexe B

## Questionnaire utilisé pour l'expérimentation

### B.1 Informations sur le participant

1. Nom du participant
2. Âge
3. Situation familiale
4. Situation professionnelle
5. Est-ce que vous vous vivez seul(e)?
6. Utilisez-vous souvent un ordinateur?

### B.2 Habitudes de vie

1. Quels sont les moyens que vous utilisez pour vous rappeler de vos rendez-vous importants (agenda, calendrier, bloc-notes, rien...)?
2. Dans votre vie quotidienne, qu'utilisez-vous comme moyen de rappel ?
  - Agenda
  - Calendrier
  - Emploi du temps

### B.3. UTILISATION DE PRADHA

- GoogleAgenda<sup>®</sup>
  - Facebook<sup>®</sup>
  - iCal<sup>®</sup>
  - Rien du tout
3. Pourquoi? Pour quelles activités?
  4. Est-ce que votre agenda est électronique?
  5. Est-il mobile?
  6. Comment utilisez-vous votre agenda? Mélangez-vous activités personnelles et professionnelles?
  7. Est-ce que vous avez un calendrier fixe à la maison? Si oui où est-il placé?  
Emplacement:
  8. Est-ce que vous partagez votre calendrier avec plusieurs personnes?
  9. Quel(s) moyen(s) utilisez-vous pour vous rappeler des choses à faire (liste de commissions... ). Qu'est-ce qui est le plus utile dans l'agenda ou le moyen de rappel que vous utilisez?
  10. Qu'est-ce que vous aimeriez changer dans l'agenda ou le moyen de rappel que vous utilisez? Quelle amélioration vous serait utile?

### **B.3 Utilisation de PRADHA**

1. Pendant l'expérimentation, est ce que vous avez organisé votre emploi du temps différemment de ce que vous pensiez au début?
2. Est ce que vous avez modifié l'ordre de réalisation des activités? Pourquoi?
3. Avez-vous vu les rappels visuels d'Amelis? Vous ont-ils paru utiles?
4. Est ce que vous avez compris dans quelles situations vous avez reçu certains rappels sonores? Pourquoi?
5. Quel type trouviez-vous le plus efficace?

#### B.4. APPRÉCIATION DE PRADHA

### **B.4 Appréciation de PRADHA**

1. Est ce que vous avez aimé votre expérience?
2. Est-ce que certains rappels vous ont paru trop insistants? Désagréables? Si oui, lesquels?
3. Dans votre vie courante, y aurait-il des situations chez vous où vous aimeriez avoir des rappels en fonction de ce que vous faites?
4. Est-ce que vous aimeriez avoir un système comme ça chez vous?
5. Connaissez vous des personnes pour qui ce système serait utile?
6. Autres commentaires?

# Annexe C

## Résultats des expérimentations

Ci-dessous sont représentés une partie du détail des résultats des expérimentations. Le tableau C.1 présente les moyens de rappels utilisés pour réaliser les activités. Ces données ont été générées à partir des données des caméras, combinées aux fichiers log des différents programmes. Le tableau C.2 montre les résultats détaillés des deux algorithmes : mixture et simple association (simple matching dans le tableau). Y sont présentés également les différents niveaux de priorité tels qu'estimés par les participants. Le tableau C.3 indique les résultats de reconnaissance et de précision de chacun des algorithmes. Dans tous ces tableaux, les lignes surlignées représentent les activités reconnaissables.

Heure prévue	Activité	Réalisation		Temps de réalisation					Prise en compte du signal par le participant					Rappels émis			
		Pas réalisée	partielle	entière	Pas réalisée	en avance	à l'heure	en retard	Consultation agenda	Réveil	Ameljs visuel	1 <sup>er</sup> Rappel	2 <sup>me</sup> Rappel	Sans rappel	Rappel 1		Rappel 2
															Rappel 1	Rappel 2	
9:00:00	Se réveiller	0	0	12	0	0	12	0	0	0	12	0	0	0	0	0	0
9:03:00	Aller aux toilettes	0	0	12	0	0	12	0	8	0	1	0	0	3	2	1	0
9:04:00	Shower	1	0	11	1	8	3	0	4	0	0	0	0	6	0	0	0
9:07:00	Ranger la chambre	0	0	12	0	9	3	0	3	0	0	0	0	9	0	0	0
9:08:00	Préparer le déjeuner	0	0	12	0	10	2	0	7	0	1	1	0	4	9	4	0
9:15:00	Manger	0	0	12	0	4	3	5	2	0	0	2	4	4	10	9	0
9:20:00	Ranger toute la table de la salle à manger	0	1	11	0	3	2	7	8	0	0	0	0	4	0	0	0
9:21:00	Arriver au niveau 7 de travail	4	6	2	4	1	3	4	7	0	0	0	0	2	0	0	0
9:23:00	Faire la vaisselle	0	0	12	0	1	8	3	4	0	0	8	1	1	9	6	0
9:28:00	Se brosser les dents	0	0	12	0	3	8	1	7	0	2	0	0	3	0	0	0
9:30:00	Prendre ses médicaments	0	0	12	0	2	8	2	2	0	0	2	1	7	10	7	0
9:45:00	Se déshabiller dans le salon	4	1	7	4	3	1	4	7	0	1	0	0	0	0	0	0
9:51:00	Téléphoner à Pierre-Yves	0	2	10	0	4	6	2	9	0	2	0	0	1	0	0	0
9:40:00	Faire le repassage	0	2	10	0	2	4	6	4	0	1	2	4	2	10	8	0
9:50:00	Aller aux toilettes	1	0	11	1	1	10	0	3	0	0	9	0	0	11	3	0
9:52:00	Prendre l'autobus	0	0	12	0	2	5	5	3	0	3	0	0	5	0	0	0
Total reconnaissance		10	12	170	10	65	78	39	78	12	11	24	10	51	61	38	0
Totaux (%)		5,21%	6,25%	88,54%	5,21%	33,85%	40,63%	20,31%	41,94%	6,45%	5,91%	12,90%	5,38%	27,42%	27,42%	0	0

figure C.1 – Grille présentant les résultats de reconnaissance et les réalisations des activités

Heure prévue	Activité	1ère fois												2ème fois						Rappel justifié trop insistant				Rappel supplémentaire nécessaire						
		Simple matching				Mixture				Simple matching				Mixture		Mixture		NB priority 0	Nb priority 1	Nb priority 2	Nb priority 3									
		TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	TP	TN					FP	FN							
9:00:00	Se réveiller	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	34	0	0	0	0	2	10
9:03:00	Aller aux toilettes	12	0	0	0	10	0	0	2	1	1	0	0	1	0	0	0	0	0	0	0	0	0	13	5	3	2	2	2	2
9:04:00	Shabiller	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	14	2	6	4	0	0	0
9:07:00	Ranger la chambre	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	19	2	3	5	2	2	2
9:08:00	Préparer le déjeuner	7	2	0	3	3	0	0	4	7	0	0	2	5	0	0	2	2	0	9	5	2	0	9	5	5	2	0	0	0
9:15:00	Manger	7	4	0	1	2	3	0	2	3	5	0	2	1	0	0	2	3	0	11	6	2	3	11	6	2	3	1	1	1
9:20:00	Ranger toute la table de la salle à manger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	16	3	2	16	3	2	7	0	0	0
9:21:00	Arriver au niveau 7, de textx	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	10	2	2	0	0	0	0
9:23:00	Faire la vaisselle	3	9	0	0	1	0	0	2	0	6	1	0	2	3	0	0	3	2	0	0	0	0	16	4	2	4	2	2	2
9:28:00	Se brosser les dents	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	13	4	5	13	4	5	1	2	2	2
9:30:00	Prendre ses médicaments	5	7	0	0	1	1	1	2	8	2	0	0	0	0	0	0	3	1	0	0	4	2	31	0	1	3	8	8	8
9:45:00	Se détendre dans le salon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	11	1	1	0	0	0	0
9:31:00	Téléphoner à Pierre-Yves	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	22	1	2	22	1	2	7	2	2	2
9:40:00	Faire le repassage	3	9	0	0	2	0	0	1	3	6	0	1	2	0	0	1	1	1	1	11	5	4	11	5	4	2	1	1	1
9:50:00	Aller aux toilettes	1	11	0	0	1	0	0	0	0	9	1	0	1	8	0	0	1	1	0	12	6	2	12	6	2	2	2	2	2
9:52:00	Prendre l'autobus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	32	0	1	6	0	1	2	2	9	9
Total reconnaissance		38	42	0	4	20	4	3	11	37	16	0	8	23	1	0	13	1	0	6	32	0	1	6	0	1	2	2	9	9

figure C.2 – Grille présentant l'émission des rappels



Heure prévue	Activité	Totaux																											
		Simple Matching						Gaussian Mixture						General															
		TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN	TP	TN	FP	FN												
		Simple matching accuracy						Mixture Gaussian accuracy						General accuracy		Simple matching FP+FN		Mixture Gaussian FP+FN											
900000	Se réveiller	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
903000	Aller aux toilettes	13	1	0	0	11	0	0	0	2	85,71%	84,62%	100,00%	11	14	2	0	11	1	0	2	0	0	0	0	0	0	0	0
904000	Shabiller	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
907000	Ranger la chambre	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
908000	Préparer le déjeuner	14	2	0	5	8	0	0	0	6	47,62%	57,14%	76,19%	8	16	6	5	8	2	0	11	5	8	2	0	3	12	0	7
915000	Manger	10	9	0	3	3	0	4	68,18%	60,00%	86,36%	6	19	4	3	3	3	3	12	0	7	4	3	3	0	0	0	0	0
921000	Ranger toute la table de la salle à manger	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
921000	Arriver au niveau 7 de triplex	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
923000	Faire la vaisselle	9	10	0	2	4	0	2	3	66,67%	44,44%	90,48%	4	19	5	2	4	10	2	5	2	4	10	2	5	2	5	5	
928000	Se brosser les dents	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
930000	Prendre ses médicaments	13	9	0	0	4	2	1	6	68,18%	46,15%	100,00%	6	22	7	0	4	11	1	6	0	4	11	1	0	4	11	1	
945000	Se dédramatiser dans le salon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
931000	Téléphoner à Pierre-Yves	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
940000	Faire le repassage	6	15	0	1	4	0	0	2	86,36%	66,67%	95,45%	4	21	2	1	4	15	0	3	1	4	15	0	0	1	15	0	
950000	Aller aux toilettes	10	12	0	1	9	0	0	1	91,30%	90,00%	95,65%	9	22	1	1	9	12	0	2	1	9	12	0	1	9	12	0	
952000	Prendre l'autobus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Total	75	58	0	12	43	5	3	24	73,10%	64,00%	91,72%	48	133	27	12	43	63	3	36	12	43	63	3	29,66%	43,45%	2,07%	24,83%	

figure C.3 – Grille présentant le décompte des reconnaissances

# Bibliographie

- [1] Jérémy BAUCHET.  
« *Archipel: un framework objet pour une approche ubiquitaire de l'assistance cognitive* ».  
Thèse de doctorat, Université de Sherbrooke (Canada), décembre 2008.
- [2] Kent BECK et Cynthia ANDRES.  
*Extreme programming explained: embrace change*.  
Addison-Wesley Professional, 2004.
- [3] Genevieve BELL et Paul DOURISH.  
« Yesterday's tomorrows: notes on ubiquitous computing's dominant vision ».  
*Personal and Ubiquitous Computing*, 11(2):133–143, 2007.
- [4] Bruno BOUCHARD, Sylvain GIROUX et Abdenour BOUZOUANE.  
« A smart home agent for plan recognition ».  
Dans *Advances in Artificial Intelligence*, pages 25–36. Springer, 2006.
- [5] Liming CHEN, Jesse HOEY, Chris D NUGENT, Diane J COOK et Zhiwen YU.  
« Sensor-Based Activity Recognition ».  
*IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS*, 42:790–808, 2012.
- [6] Liming CHEN et Chris NUGENT.  
« Ontology-based activity recognition in intelligent pervasive environments ».  
*Emerald*, 5(4):410–430, 2009.
- [7] Liming CHEN, Chris D. NUGENT et Hui WANG.  
« A Knowledge-Driven Approach to Activity Recognition in Smart Homes ».  
*Knowledge and Data Engineering, IEEE Transactions on*, 24(6):961–974, june 2012.

## BIBLIOGRAPHIE

- [8] Belkacem CHIKHAOUI, Shengrui WANG et Helene PIGOT.  
« A new algorithm based on sequential pattern mining for person identification in ubiquitous environments ».  
Dans *KDD workshop on knowledge discovery from sensor data*, pages 19–28, 2010.
- [9] Belkacem CHIKHAOUI, Shengrui WANG et H el ene PIGOT.  
« A Frequent Pattern Mining Approach for ADLs Recognition in Smart Environments ».  
Dans *Advanced Information Networking and Applications (AINA). 2011 IEEE International Conference on*, pages 248–255, march 2011.
- [10] Yi CHU, Young CHOL SONG, Richard LEVINSON et Henry KAUTZ.  
« Interactive activity recognition and prompting to assist people with cognitive disabilities ».  
*Journal of Ambient Intelligence and Smart Environments*, 4(5):443–459, 2012.
- [11] Sook-Ling CHUA, Stephen MARSLAND et Hans W GUESGEN.  
« Unsupervised Learning of Human Behaviours ».  
Dans *Proc. of Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 319–324, 2011.
- [12] Barnan DAS, Chao CHEN, Adriana M SEELYE et Diane J COOK.  
« An automated prompting system for smart environments ».  
Dans *Toward Useful Services for Elderly and People with Disabilities*, pages 9–16. Springer, 2011.
- [13] Barnan DAS, Diane J COOK, Maureen SCHMITTER-EDGEcombe et Adriana M SEELYE.  
« PUCK: an automated prompting system for smart environments: toward achieving automated prompting—challenges involved ».  
*Personal and Ubiquitous Computing*, 16(7):859–873, 2012.
- [14] C eline DESCHENEaux et H el ene PIGOT.  
« Interactive Calendar to Help Maintain Social Interactions for Elderly People and People with Mild Cognitive Impairments ».  
Dans *Ambient Assistive Health and Wellness Management in the Heart of the City*, pages 117–124. Springer, 2009.

## BIBLIOGRAPHIE

- [15] T.V. DUONG, H.H. BUI, D.Q. PHUNG et S. VENKATESH.  
« Activity recognition and abnormality detection with the switching hidden semi-Markov model ».  
Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845 vol. 1, june 2005.
- [16] Sylvain GIROUX, Jérémy BAUCHET, Hélène PIGOT, Dany LUSSIER-DESROCHERS et Yves LACHAPPELLE.  
« Pervasive behavior tracking for cognitive assistance ».  
Dans *Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*, pages 86–93. ACM, 2008.
- [17] Sylvain GIROUX et Simon GUERTIN.  
« A pervasive reminder system for smart homes ».  
*Assistive Technology Research Series*, 14:98–106, 2004.
- [18] Charles GOUIN-VALLERAND.  
« *Gestion et déploiement d'applications en milieux ubiquistes à l'aide d'OSGi* ».  
Thèse de doctorat, Université de Sherbrooke (Canada), 2008.
- [19] Charles GOUIN-VALLERAND et Sylvain GIROUX.  
« Managing and Deployment of Applications with OSGi in the Context of Smart Home ».  
Dans *Wireless and Mobile Computing, Networking and Communications, 2007. WiMOB 2007. Third IEEE International Conference on*, pages 70–78, 2007.
- [20] Thomas R. GRUBER.  
« A translation approach to portable ontology specifications ».  
*Knowledge acquisition*, 5(2):199–220, 1993.
- [21] Tao GU, Zhanqing WU, Xianping TAO, Hung Keng PUNG et Jian LU.  
« epSICAR: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition ».  
Dans *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*, pages 1–9. IEEE, 2009.
- [22] John A HARTIGAN et Manchek A WONG.  
« Algorithm AS 136: A k-means clustering algorithm ».

## BIBLIOGRAPHIE

- Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.
- [23] Geoffrey E HINTON, Michael REVOW et Peter DAYAN.  
« Recognizing handwritten digits using mixtures of linear models ».  
*Advances in neural information processing systems*, pages 1015–1022, 1995.
- [24] Melanie HOOVER et Michelle ROTERMANN.  
« Le recours aux soins à domicile par les personnes âgées et les besoins insatisfaits ».  
*Statistique Canada, no 82-003-XPF au catalogue Rapports sur la santé*, 23:3–9, décembre 2012.
- [25] Tâm HUỠNH, Ulf BLANKE et Bernt SCHIELE.  
« Scalable recognition of daily activities with wearable sensors ».  
Dans *Location-and context-awareness*, pages 50–67. Springer, 2007.
- [26] Javier Ortiz LAGUNA, Angel García OLAYA et Daniel BORRAJO.  
« A dynamic sliding window approach for activity recognition ».  
Dans *User Modeling, Adaption and Personalization*, pages 219–230. Springer, 2011.
- [27] Richard LEVINSON.  
« The planning and execution assistant and trainer (PEAT) ».  
*The Journal of Head Trauma Rehabilitation*, 12(2):85–91, 1997.
- [28] Laurent MARTEL, Eric CARON-MALENFANT, Samuel VÉZINA et Alain BÉLANGER.  
« Projections de la population active au Canada, 2006-2031 ».  
*L'Observateur économique canadien*, 20(6):3–1, 2007.
- [29] Alex MIHAILIDIS, Jennifer N BOGER, Tammy CRAIG et Jesse HOEY.  
« The COACH prompting system to assist older adults with dementia through handwashing: An efficacy study ».  
*BMC geriatrics*, 8(1):28, 2008.
- [30] Kenneth S MILLER.  
*Multidimensional gaussian distributions*.  
Wiley New York, 1964.

## BIBLIOGRAPHIE

- [31] E. NAZERFARD, P. RASHIDI et D.J. COOK.  
« Discovering Temporal Features and Relations of Activity Patterns ».  
Dans *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 1069–1075, dec. 2010.
- [32] Nam Thanh NGUYEN, Dinh Q PHUNG, Svetha VENKATESH et Hung BUI.  
« Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model ».  
Dans *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 955–960. IEEE, 2005.
- [33] Nikhil R PAL et James C BEZDEK.  
« On cluster validity for the fuzzy c-means model ».  
*Fuzzy Systems, IEEE Transactions on*, 3(3):370–379, 1995.
- [34] Maja PANTIC, Alex PENTLAND, Anton NIJHOLT et Thomas S HUANG.  
« Human computing and machine understanding of human behavior: a survey ».  
Dans *Artificial Intelligence for Human Computing*, pages 47–71. Springer, 2007.
- [35] H el ene PIGOT, Andr e MAYERS et Sylvain GIROUX.  
« The intelligent habitat and everyday life activity support ».  
Dans *5th international conference on Simulations in Biomedicine*, pages 507–516, 2003.
- [36] Martha E. POLLACK.  
« Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment ».  
*AI magazine*, 26(2):9, 2005.
- [37] Martha E. POLLACK, Laura BROWN, Dirk COLBRY, Colleen E MCCARTHY, Cheryl OROSZ, Bart PEINTNER, Sailesh RAMAKRISHNAN et Ioannis TSAMARDINOS.  
« Autominder: An intelligent cognitive orthotic system for people with memory impairment ».  
*Robotics and Autonomous Systems*, 44(3):273–282, 2003.
- [38] Martha E. POLLACK, Laura BROWN, Dirk COLBRY, Cheryl OROSZ, Bart PEINTNER, Sailesh RAMAKRISHNAN, Sandra ENGBERG, Judith T. MATTHEWS, Jacqueline DUNBAR-JACOB, Colleen E. MCCARTHY, Sebastian THRUN, Michael

## BIBLIOGRAPHIE

- MONTEMERLO, Joelle PINEAU et Nicholas ROY.  
« Pearl: A mobile robotic assistant for the elderly ».  
Dans *AAAI workshop on automation as eldercare*, volume 2002, pages 85–91, 2002.
- [39] Martha E. POLLACK, Colleen E MCCARTHY, Sailesh RAMAKRISHNAN, Ioannis TSAMARDINOS, Laura BROWN, Steve CARRION, Dirk COLBRY, Cheryl OROSZ et Bart PEINTNER.  
« Autominder: A planning, monitoring, and reminding assistive agent ».  
Dans *7th International Conference on Intelligent Autonomous Systems*, pages 265–272, 2002.
- [40] Youcef RAHAL, Philippe MABILLEAU et Helene PIGOT.  
« Bayesian filtering and anonymous sensors for localization in a smart home ».  
Dans *Advanced Information Networking and Applications Workshops, 2007, AINAW'07. 21st International Conference on*, volume 2, pages 793–797. IEEE, 2007.
- [41] Richard A REDNER et Homer F WALKER.  
« Mixture densities, maximum likelihood and the EM algorithm ».  
*SIAM review*, 26(2):195–239, 1984.
- [42] Douglas A REYNOLDS et Richard C ROSE.  
« Robust text-independent speaker identification using Gaussian mixture speaker models ».  
*Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.
- [43] Daniele RIBONI, Linda PARESCHI, Laura RADAELLI et Claudio BETTINIM.  
« Is ontology-based activity recognition really effective? ».  
Dans *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 427–431, march 2011.
- [44] Jehad SARKAR, Young-Koo LEE et Sungyoung LEE.  
« A Smoothed Naive Bayes-Based Classifier for Activity Recognition ».  
*IETE Tech Rev*, 27:107–119, février 2010.
- [45] Richard SIMPSON, Debra SCHRECKENGHOST, Edmund F LOPRESTI et Ned KIRSCH.

## BIBLIOGRAPHIE

- « Plans and planning in smart homes ».  
Dans *Designing Smart Homes*, pages 71–84. Springer, 2006.
- [46] Pang-Ning TAN, Michael STEINBACH et Vipin KUMAR.  
*Introduction to Data Mining, (First Edition)*.  
Addison-Wesley Longman Publishing Co., Inc., 2005.
- [47] Emmanuel TAPIA, Stephen INTILLE et Kent LARSON.  
« Activity Recognition in the Home Using Simple and Ubiquitous Sensors ».  
Dans Alois FERSCHA et Friedemann MATTERN, éditeurs, *Pervasive Computing*,  
volume 3001 de *Lecture Notes in Computer Science*, pages 158–175. Springer  
Berlin / Heidelberg, 2004.  
10.1007/978-3-540-24646-6\_10.
- [48] Tim VAN KASTEREN, Athanasios NOULAS, Gwenn ENGLEBIENNE et Ben  
KRÖSE.  
« Accurate activity recognition in a home setting ».  
Dans *Proceedings of the 10th international conference on Ubiquitous computing*,  
UbiComp '08, pages 1–9. ACM, 2008.
- [49] TLM van KASTEREN, AK NOULAS et BJA KRÖSE.  
« Conditional Random Fields versus Hidden Markov Models for activity recog-  
nition in temporal sensor data ».  
*Proceedings of the 14th Annual Conference of the Advanced School for Computing  
and Imaging*, pages 1–7, 2008.
- [50] Xizhao WANG, Yadong WANG et Lijuan WANG.  
« Improving Fuzzy C-Means Clustering based on Feature-Weight Learning ».  
*Pattern Recognition Letters*, 25(10):1123–1132, 2004.
- [51] Mark WEISER.  
« The computer for the 21st century ».  
*Scientific american*, 265(3):94–104, 1991.
- [52] Danny WYATT, Matthai PHILIPOSE et Tanzeem CHOUDHURY.  
« Unsupervised activity recognition using automatically mined common sense ».  
Dans *Proceedings of the National Conference on Artificial Intelligence*, vo-



## BIBLIOGRAPHIE

lume 20, page 21. Menlo Park, CA; Cambridge, MA; London; AAI Press; MIT Press; 1999, 2005.