

Vers une méthode d'acquisition et d'analyse de données pour le dépistage
précoce de la maladie d'Alzheimer dans un environnement intelligent

par

Mathieu Gagnon

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, 16 janvier 2018

Le 16 janvier 2018

le jury a accepté le mémoire de Monsieur Mathieu Gagnon dans sa version finale.

Membres du jury

Professeur Sylvain Giroux
Directeur de recherche
Département d'informatique

Professeur Hélène Pigot
Membre interne
Département d'informatique

Professeur André Mayers
Président-rapporteur
Département d'informatique

Sommaire

Dans les pays développés comme le Canada, on remarque de plus en plus un vieillissement de la population et parallèlement une augmentation du nombre de personnes atteintes de déficiences cognitives. La cause la plus répandue est la démence de type Alzheimer (DTA), aussi connue sous le nom de maladie d'Alzheimer. Les incapacités cognitives causées par la DTA entraînent des difficultés dans les Activités Instrumentales de la Vie Quotidienne (AIVQ). Certains chercheurs considèrent qu'il existe des marqueurs cognitifs propres à la DTA, c'est-à-dire que les personnes atteintes peuvent présenter des difficultés cognitives observables lors de la réalisation de tâches complexes bien des années avant son diagnostic. Un habitat intelligent, muni de capteurs de mouvements, débitmètres, capteurs de contacts pour portes et tiroirs, permet de mesurer divers aspects de la performance dans la vie quotidienne lors de la réalisation d'une AIVQ.

Partant de ce constat, nos travaux explorent comment des habitats intelligents pourraient permettre de détecter la DTA de façon précoce. Dans cette optique, nous proposons une méthodologie expérimentale rigoureuse. Tout d'abord, nous avons conçu, implémenté et déployé un système d'acquisition de données hétérogènes fiable intégrant réseaux de capteurs variés, vidéos de l'expérimentation et annotations de l'expérimentateur. Ensuite, nous avons exploré divers algorithmes de classification pour distinguer trois catégories de participants : sans troubles cognitifs, avec troubles cognitifs légers et avec DTA. Bien que l'acquisition des données soit complexifiée par la multiplication des sources, notre approche permet la validation des données acquises. Cet aspect est important car la qualité de ces données, acquises lors des expérimentations, influence grandement la performance des algorithmes de classification. Finalement, ce projet étudie comment réaliser et comparer les données et les résultats d'expérimentations menées sur des sites différents en termes de configuration spatiale, de densité et de positionnement des capteurs. C'est pourquoi des expérimentations avec des personnes âgées se sont déroulées aux laboratoires DOMUS (Sherbrooke) et du CRIUGM (Montréal). Les expérimentations ont utilisé le même protocole d'expérimentation où des

participants ont eu un temps déterminé pour réaliser la même liste de tâches. Les participants ont été recrutés et séparés en trois groupes selon leur diagnostic : sujets sains, sujets atteints d'un trouble cognitif léger (TCL) et sujets atteints de démence de type Alzheimer (DTA). Les données recueillies ont ensuite été annotées et traitées en vue d'une analyse à l'aide de techniques d'apprentissage automatique. Une première approche de classification par arbre de décision simple a permis d'observer une différence significative entre les données d'expérimentation des personnes saines et celles des personnes TCL. Par contre, aucune différence claire n'est apparue entre les personnes DTA et les autres catégories. En conclusion, d'autres représentations de données et d'autres algorithmes sont toujours en cours d'exploration par d'autres membres de notre équipe. Les résultats préliminaires semblent prometteurs.

Mots-clés : Dépistage précoce de la maladie d'Alzheimer. Activité Instrumentale de la Vie Quotidienne. Intelligence ambiante. Habitat intelligent. Réseau de capteurs. Annotation de données. Classification. Expérimentation multisite.

Remerciements

Je désire remercier mon directeur Sylvain Giroux qui a toujours pris le temps de répondre rapidement à mes courriels les plus urgents ... (je blague !). Non, pour vrai, mon directeur m'a été d'un grand soutien tout au long de ma maîtrise; je manque de mots pour le remercier comme il se doit mais je peux dire que c'est une personne facile d'approche, joviale, à l'écoute et très vive d'esprit. Je me souviendrai longtemps des méchouis, pots de départ et dîners au Caméléon, à la taverne Alexandre ou au Café Aragon.

J'en profite aussi pour remercier Hélène Pigot, codirectrice du laboratoire DOMUS, qui m'a aussi été d'un grand soutien au cours de ma maîtrise. Les exemples ne me viennent pas à l'esprit, mais sa franchise et sa manière de penser complétait généralement très bien les conseils de mon directeur lorsque j'en avais besoin. Je lui souhaite tout le succès qu'elle mérite dans ses projets futurs!

Je voudrais remercier mon acolyte Jules Randolph, qui a travaillé si dur pour créer de ses propres mains l'application CoLab, pour régler les 12 000 problèmes que nous avons avec les caméras IP et les capteurs, et simplement, je le remercie pour tout le soutien qu'il m'a offert durant ces trois longues (!) années.

Je voudrais remercier mes collègues de Montréal : Nathalie Bier, pour avoir chapeauté le projet DEI d'une main de maître; Christine Ménard, pour s'être si bien occupée du recrutement et des expérimentations (ce fut un plaisir travailler avec toi!); Maxime Lussier pour sa motivation et ses idées pour l'analyse des données cliniques et de capteurs du projet et Belkacem Chikhaoui pour son aide sur l'analyse des données de capteurs, ainsi que tous les participants au projet.

Je voudrais remercier mes partenaires de babyfoot, de sorties et de folies régulières : Frédéric Bergeron, Robert Radziszewski, Nicola Kuijpers, Hubert Ngankam, Wathek Bellah Loued, Léopold Lieb, Damien Lockner, Yannick Drolet, Camila Oliveira et Amandine Porcher, qui ont su agrémente mes journées de tant de fous rires et de bons temps, et d'idées aussi!

Je voudrais remercier « mon stagiaire » Cyril Pastourel, qui m'a été d'une aide cruciale lors des premières étapes du projet, à l'été 2014. Il était d'une productivité surprenante et m'a appris tant de trucs cool de la vie de Français (comme le verlan). Je désire remercier aussi Jérôme Bournaud pour d'autres trucs cool de la vie de Français, mais aussi pour du soutien au niveau des capteurs. Sans oublier Antoine Croteau et Keven Fecteau qui ont élaboré la première version de l'Application Projet Dépistage, l'application tablette utilisée lors des expérimentations.

Je voudrais remercier Marc Chevalaz et Christina Khnaisser pour les sorties organisées intégrant les nouveaux stagiaires et d'autres amis gravitant autour du DOMUS, et pour l'aide apportée à plusieurs moments du projet.

Je désire aussi remercier les nombreux stagiaires que je n'ai pas mentionnés qui sont passés au DOMUS, agrémentant la vie au labo et laissant leur marque dans ma mémoire. Je ne veux pas oublier non plus mes autres collègues du DOMUS, professionnels et anciens étudiants, qui auront marqué mon séjour au DOMUS, même si je ne vous ai pas cité personnellement (ça serait trop long!).

Je désire finalement remercier ma conjointe Jessica Morin-Côté, mes parents et mes frères, qui m'ont soutenu tout le long de ce long périple rempli d'obstacles, et mes amis de longue date, dont Simon Roy, qui m'ont encouragé à faire la maîtrise.

Table des matières

Sommaire	iii
Remerciements.....	v
Table des matières	vii
Liste des abréviations.....	xi
Liste des tableaux.....	xii
Liste des figures	xiii
Introduction.....	1
Contexte	1
Objectifs.....	3
Méthodologie	4
Contributions.....	9
Structure du mémoire.....	11
Chapitre 1 État de l’art.....	13
1.1 La maladie d’Alzheimer.....	13
1.1.1 Dépistage de la maladie d’Alzheimer	14
1.2 Expérimentations au sein d’environnements intelligents.....	15
1.2.1 Détection autonome et non intrusive d’un TCL chez les personnes âgées... 15	
1.2.2 Évaluation autonome de la santé cognitive en utilisant la domotique.....	16
1.3 Conclusion.....	17
Chapitre 2 Protocole d’expérimentation.....	19
2.1 Scénario d’expérimentation.....	19
2.1.1 Tests cognitifs	19
2.1.2 Tâches observées	21

2.2	Déroulement des expérimentations	22
2.2.1	Conditions initiales d'expérimentation	22
2.2.2	Lors d'une expérimentation	24
2.2.3	À la suite d'une expérimentation	25
2.3	Automatisation des vérifications technologiques.....	26
2.4	Comparaison des sites	27
2.5	Conclusion.....	27
Chapitre 3 Environnement et architecture		29
3.1	Environnement	29
3.1.1	Réseau de capteurs filaires.....	31
3.1.2	Réseau de capteurs sans fil	33
3.1.3	Multimédia.....	37
3.2	Architecture.....	39
Chapitre 4 Annotation des données		42
4.1	Application Projet Dépistage (APD).....	42
4.1.1	But de l'application.....	42
4.1.2	Fonctionnalités.....	43
4.1.3	Stockage des données	48
4.1.4	Perspectives.....	49
4.2	CoLab	49
4.2.1	But de l'application.....	49
4.2.2	Fonctionnalités.....	50
4.2.3	Perspectives.....	54
4.3	Conclusion.....	55
Chapitre 5 Expérimentations et prétraitement des données		56
5.1	Informations démographiques des participants.....	56
5.2	Déroulement et problèmes lors des expérimentations.....	57
5.3	Données récoltées.....	59

5.3.1	Description des données	59
5.3.2	Prétraitement des données.....	62
5.4	Conclusion.....	68
Chapitre 6 Analyse.....		69
6.1	Représentation des données	69
6.1.1	Nombre de déclenchements des capteurs	70
6.1.2	Temps passé dans chaque pièce et durée d'utilisation.....	72
6.2	Classification automatique à l'aide des capteurs filaires	73
6.2.1	Algorithmes utilisés	74
6.2.2	Premiers résultats	79
6.2.3	Fiabilité des résultats avec OneR.....	81
6.2.4	Capteurs les plus discriminants.....	83
6.2.5	Analyse à deux classes : sain ou avec troubles cognitifs	86
6.2.6	Classification avec fenêtres temporelles.....	91
6.2.7	Conclusion	91
6.3	Analyse statistique à l'aide des capteurs sans fil.....	92
6.4	Faisabilité d'une expérimentation multisite	93
Conclusion		95
	Critique du travail	96
	Travaux futurs de recherche.....	96
	Perspective	98
Bibliographie.....		99
Annexe A Capteurs sans fil		106
A.1	Liste des capteurs	106
A.1.1	Convention pour les noms des capteurs sans fil.....	106
A.1.2	Capteurs de contact électromagnétique.....	107
A.1.3	Capteurs de mouvement	109
A.1.4	Capteurs de consommation électrique.....	110

A.2	Sous-ensemble de capteurs nécessaires pour les expérimentations du projet DEI .	110
A.2.1	Capteurs de contact électromagnétique.....	110
A.2.2	Capteurs de mouvement.....	111
A.2.3	Capteurs de consommation électrique.....	112
Annexe B	Listes de vérification lors des expérimentations	113
B.1	Liste préexpérimentation.....	113
B.2	Liste post-expérimentation.....	116
Annexe C	Scripts utilisés dans le projet	119
C.1	Copier la base de données SQLite interne d'Android.....	119
C.2	Concaténer les données en un seul fichier (concatenate_exp).....	120
C.3	Transformer les décimaux à virgule en décimaux à point	120
C.4	Binariser les valeurs des débitmètres	121
Annexe D	Critères de sélection des participants	123
D.1	Participants sans troubles cognitifs	123
D.2	Participants avec TCL	124
D.3	Participants avec DTA	124

Liste des abréviations

CASAS	<i>Center for Advanced Studies in Adaptive Systems</i>
CRIM	Centre de Recherche Informatique de Montréal
CRIUGM	Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal
DI	Département d'informatique
DOMUS	DOMotique et informatique Mobile à l'Université de Sherbrooke
ORCATECH	<i>ORegon Centre for Aging and TECHnology</i>
UdeS	Université de Sherbrooke
AIVQ	Activités Instrumentales de la Vie Quotidienne
DTA	Démence de Type Alzheimer
TCL	Trouble Cognitif Léger, en anglais <i>Mild Cognitive Impairment</i> , MCI
APD	Application Projet Dépistage
API	<i>Application Programming Interface</i> , en français Interface de programmation applicative
BD	Base de données
CSV	<i>Comma-Separated Values</i>
FTP	<i>File Transfer Protocol</i>
Go	Gigaoctet
HTTP	<i>HyperText Transfer Protocol</i>
IP	<i>Internet Protocol</i>
Mo	Mégaoctet
RTSP	<i>Real Time Streaming Protocol</i>
UCT	Unité Centrale de Traitement, en anglais <i>Central Processing Unit</i> , CPU

Liste des tableaux

Tableau 1 – Tests cognitifs standardisés administrés à tous les participants.....	20
Tableau 2 – Tests et questions supplémentaires administrés aux participants TCL seulement	20
Tableau 3 – Différences entre les laboratoires DOMUS et du CRIUGM	27
Tableau 4 – Exemple de données du réseau de capteurs filaires	59
Tableau 5 – Exemple de données du réseau de capteurs sans fil.....	60
Tableau 6 – Exemple de données d’annotation de la durée des tâches	61
Tableau 7 – Exemple de données d’heures de début et de fin d’expérimentation.....	62
Tableau 8 – Exemple de données représentées en nombre de déclenchements	71
Tableau 9 – Exemple de données représentées en temps passé dans la pièce	72
Tableau 10 – Premiers résultats de l’apprentissage supervisé avec 3 classes (Sain, DTA et TCL) en utilisant tous les capteurs.....	80
Tableau 11 – Tests pour connaître dans quels cas le capteur MV1 se déclenche.....	82
Tableau 12 – Résultats de l’apprentissage supervisé avec 3 classes (Sain, DTA, TCL) en utilisant un ensemble réduit d’attributs.....	85
Tableau 13 – Résultats de l’apprentissage supervisé sur les classes Sain et Avec_TC en utilisant tous les capteurs	87
Tableau 14 – Résultats de l’apprentissage supervisé avec 2 classes (Sain et TCL) en utilisant tous les capteurs	88
Tableau 15 – Résultats de l’apprentissage supervisé avec les classes Sain et Avec_TC en utilisant le sous-ensemble des cinq capteurs les plus discriminants.....	89
Tableau 16 – Résultats de l’apprentissage supervisé avec les classes Sain et TCL en utilisant le sous-ensemble des cinq capteurs les plus discriminants.....	90

Liste des figures

Figure 1 – Méthodologie de recherche utilisée.....	4
Figure 2 – Plan du laboratoire DOMUS	30
Figure 3 – Plan du laboratoire du CRIUGM.....	31
Figure 4 – Plan de sol officiel du laboratoire DOMUS	32
Figure 5 – Élévations des armoires et meubles de l’appartement du laboratoire DOMUS	32
Figure 6 – Emplacement des capteurs de contact et de consommation électrique	34
Figure 7 – Emplacement des capteurs de mouvement sans fil	35
Figure 8 – Plan du laboratoire du CRIUGM.....	36
Figure 9 – Emplacement des caméras fixes	38
Figure 10 – Architecture du système d’acquisition de données	39
Figure 11 – Écran de démarrage avec affichage du clavier Android.....	44
Figure 12 – Écran principal avec sous-tâches dans divers états	45
Figure 13 – Écran principal avec une nouvelle sous-tâche ajoutée	46
Figure 14 – Écran d’évaluation des tâches avec la tâche « Préparer un café » sélectionnée ...	47
Figure 15 – Fenêtre demandant d’attribuer une note à la tâche « Préparer un café » que l’on vient de sélectionner	47
Figure 16 – Choix d’une expérimentation du projet DEI	51
Figure 17 – Écran principal de CoLab.....	52
Figure 18 – Détails d’une tâche dans l’explorateur temporel	52
Figure 19 – Liste de tâches et choix des couleurs d’affichage	53
Figure 20 – Modification d’une tâche.....	53
Figure 21 – Grille d’analyse comportementale utilisée par les spécialistes	54
Figure 22 – Maquette du design proposé de CoLab	55
Figure 23 – Technique utilisée pour représenter les données en un seul vecteur par participant	70
Figure 24 – Fonctionnement de la validation croisée (source : [50])	75
Figure 25 – Théorème de Bayes	77

Figure 26 – Équation de Bayes utilisée par NaiveBayes	78
Figure 27 – Emplacement du capteur MV1	81
Figure 28 – Script qui copie la base de données SQLite interne d'Android.....	119
Figure 29 – Script pour concaténer les données des expérimentations en un seul fichier.....	120
Figure 30 – Script qui binarise les valeurs des débitmètres.....	122

Introduction

Contexte

Au Canada seulement, 564 000 personnes sont actuellement atteintes d'une maladie affectant la cognition, soit 1,5% de la population. Selon le rapport d'octobre 2016 de l'Agence de la santé publique du Canada [1], ce nombre devrait augmenter à 937 000 Canadiens atteints d'une maladie affectant la cognition dans 15 ans. Aussi, plus de 10,4 milliards de dollars sont dépensés annuellement par les Canadiens pour prendre soin des personnes atteintes de maladies affectant la cognition. Ainsi, il est important de détecter ces maladies le plus rapidement possible pour ensuite mettre en place des solutions visant à retarder l'apparition des symptômes, tout en favorisant le maintien à domicile.

Jusqu'à maintenant, aucun marqueur biologique propre à la maladie d'Alzheimer, permettant de la différencier clairement des autres types de démences, n'a encore été découvert [2], [3]. Par contre, une accumulation de peptides de type beta amyloïde ($A\beta$) dans le cerveau est généralement un indicateur de la maladie d'Alzheimer, mais le seul moyen de voir cette accumulation hors de tout doute est lors d'une autopsie. Il est évident qu'il serait préférable de pouvoir identifier la DTA plus tôt si l'on veut prolonger la qualité de vie du patient le plus possible.

Certains auteurs considèrent qu'il existe des marqueurs cognitifs propres à la DTA, c'est-à-dire que les personnes atteintes peuvent présenter des difficultés cognitives observables lors de la réalisation de tâches complexes bien des années avant l'émission du diagnostic [4], [5]. Ces tâches complexes, plus précisément nommées activités instrumentales de la vie quotidienne (AIVQ) [6], sont des tâches de la vie courante telles que gérer son budget, préparer un repas, faire le ménage ou ranger ses vêtements.

Ainsi, une avenue possible pour le dépistage précoce de la maladie d'Alzheimer serait d'utiliser un environnement intelligent pour identifier des difficultés rencontrées lors de la réalisation d'AIVQ et de comparer les comportements de participants avec et sans troubles cognitifs, lorsqu'ils évoluent dans un environnement nouveau et complexe. Un environnement intelligent est muni de capteurs permettant d'acquérir et d'appliquer des connaissances obtenues sur lui-même et sur ses habitants [7], [8]. Dans le cas qui nous concerne, des personnes âgées pourraient venir réaliser des AIVQ lors d'expérimentations supervisées dans un appartement intelligent comme le laboratoire DOMUS [9]. Cet appartement représenterait un environnement nouveau pour elles, leur demandant ainsi une bonne capacité d'adaptation et sollicitant fortement leurs fonctions cognitives, plus que si on les observait à leur résidence. C'est donc pour évaluer la faisabilité de cette nouvelle approche pour le dépistage précoce de la DTA qu'est né un projet pilote sur le dépistage en environnement intelligent (DEI), mettant en scène les appartements intelligents du laboratoire DOMUS et du Centre de Recherche de l'Institut Universitaire de Gériatrie de Montréal (CRIUGM) [10]¹.

Pour mener à bien ce projet, un système d'acquisition de données hétérogènes est mis en place au sein des laboratoires DOMUS et du CRIUGM pour recueillir des mesures quantitatives et objectives sur la réalisation des AIVQ par le biais de capteurs. Des expérimentations se déroulent aux deux laboratoires où des participants ont un temps déterminé pour réaliser la même liste de tâches. Ces expérimentations obéissent à un scénario prédéfini où un participant accomplira les tâches demandées dans l'ordre qu'il voudra, avec la possibilité de faire du multitâches. Toutes les expérimentations sont faites en suivant le même protocole d'expérimentation. Par contre, les deux laboratoires n'ont pas la même configuration spatiale (agencement des pièces et des meubles, etc.) ni le même réseau de capteurs, mais tous deux utilisent entre autres des capteurs sans fil communiquant ensemble à l'aide de la technologie Z-Wave [11]. Au CRIUGM, nous n'utilisons comme capteurs que le strict nécessaire, donc des

¹ Les membres de l'équipe sont : Pr Nathalie Bier, erg., Ph.D. (CRIUGM), chercheure principale, Pr Sylvain Giroux et Pr Hélène Pigot (DOMUS), co-chercheurs principaux, Christine Ménard (CRIUGM), ergothérapeute et expérimentatrice principale, Dr Maxime Lussier, Psy. D., Ph. D. (CRIUGM), neuropsychologue et analyste de données, Jules Randolph (DOMUS), étudiant à la maîtrise en génie logiciel au DOMUS et moi-même.

capteurs sans fil sur les portes et les appareils qui doivent impérativement être utilisés pour accomplir les tâches demandées (par exemple, un capteur sur une porte qui doit absolument être ouverte pour trouver les filtres servant à faire le café). Ceci permet de limiter les coûts et de vérifier si un petit nombre de capteurs est suffisant pour obtenir des résultats significatifs. Des capteurs de mouvement, de contact électromagnétique et de consommation électrique sont utilisés. Au laboratoire DOMUS, un réseau similaire, constitué des mêmes modèles et types de capteurs sans fil, est installé en plus du réseau de capteurs filaires beaucoup plus complet déjà présent. Ce réseau filaire comprend des capteurs de contact, de mouvement, de luminosité, de température et de débit d'eau. Le grand nombre de capteurs assure une redondance des données et d'avoir accès à toutes les informations qu'il sera possible de tirer de ces expérimentations.

Objectifs

Le but du projet DEI est d'évaluer la faisabilité d'utiliser un environnement intelligent pour identifier des signes précoces de la DTA lors de la réalisation d'AIVQ par les personnes âgées, lorsqu'elles évoluent dans un environnement nouveau et complexe. Les objectifs de cette maîtrise sont divisés en deux catégories : les objectifs liés aux expérimentations et ceux liés à l'analyse des données.

Expérimentations

- O1. Définir une méthode standardisée multisite pour mesurer et analyser les comportements de participants;
- O2. Vérifier la faisabilité de réaliser une expérimentation multisite, en impliquant deux laboratoires intelligents (DOMUS et CRIUGM) et en vérifiant la possibilité de générer des données similaires, comparables et pouvant être composées;
- O3. Déterminer le type optimal de réseaux (p. ex. sans fil ou filaire) et la quantité optimale de dispositifs pour obtenir des résultats significatifs dans la prédiction d'un diagnostic cognitif.

Analyse des données

- O4. Documenter et comparer les comportements de participants avec et sans troubles cognitifs lors de la réalisation d'AIVQ au sein d'un environnement intelligent équipé d'appareils hétérogènes;
- O5. Prédire le diagnostic cognitif des participants à partir des données recueillies par l'environnement intelligent à l'aide d'algorithmes de classification automatique;
- O6. Déterminer des corrélations potentielles entre certaines variables observées, comme le nombre d'ouvertures de portes ou le temps requis pour accomplir les tâches, et le diagnostic cognitif des participants;

Méthodologie

La méthodologie utilisée dans le cadre de cette recherche est divisée en quatre grandes étapes: élaboration du protocole d'expérimentation, acquisition des données lors des expérimentations, annotation et prétraitement des données recueillies, analyse des données (Figure 1). Chacune de ces étapes présente des défis importants.

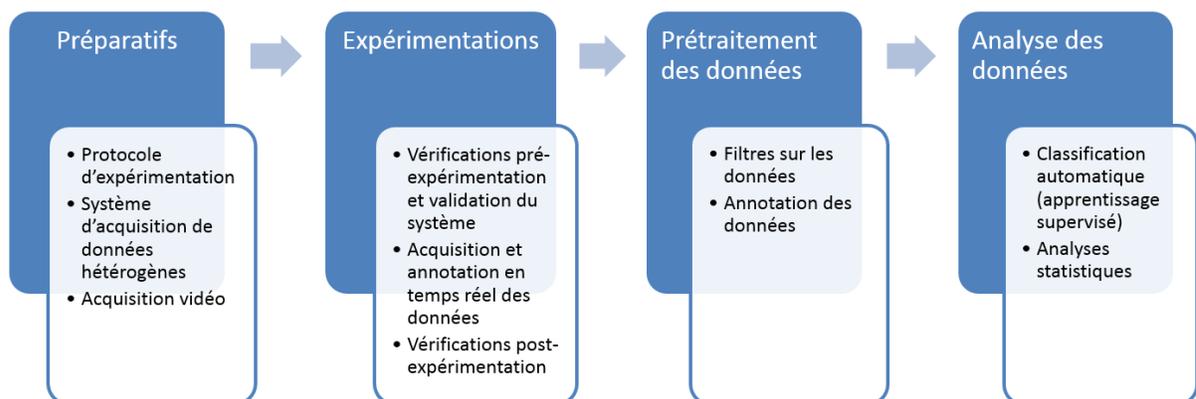


Figure 1 – Méthodologie de recherche utilisée

Élaboration du protocole d'expérimentation.

- M1. Élaboration du protocole d'expérimentation à respecter en considérant l'aspect multisite;
- M2. Conception, implémentation et déploiement du système d'acquisition de données et de l'application Android qui serviront à recueillir des mesures objectives sur la réalisation des AIVQ;
- M3. Conception, implémentation et déploiement du système d'acquisition vidéo qui servira à filmer les participants lors de la réalisation des AIVQ;

Le protocole d'expérimentation détermine les tâches à réaliser. Les tâches choisies dépendent de la configuration des lieux de l'environnement intelligent, du type et du nombre capteurs, des paramètres internes de ces capteurs, etc. Tous ces éléments influencent grandement le prétraitement et l'analyse des données acquises. Par exemple, le choix et le placement des capteurs déterminent le niveau d'analyse et de reconnaissance des tâches : « y a-t-il un ou des capteurs appropriés pour reconnaître cette tâche? », « quel est le niveau de fiabilité de ces capteurs ? », « y a-t-il une source redondante d'information pour valider les données recueillies ? », etc.

Acquisition des données lors des expérimentations.

- M4. Vérification des conditions initiales d'expérimentations, incluant la validation des systèmes, pour chacune des expérimentations;
- M5. Acquisition et annotation en temps réel des données recueillies pendant chaque expérimentation;
- M6. Vérification post-expérimentation des systèmes et état des lieux, pour chacune des expérimentations;

L'acquisition des données doit composer avec un environnement extrêmement hétérogène. Les données reçues des capteurs sont de type binaire (ouvert ou fermé, détection ou non) pour les capteurs de contact et de mouvement, ou analogique pour les capteurs de consommation électrique, de luminosité, de débit d'eau et de température. Dans le réseau de capteurs sans fil, les données sont recueillies en suivant une approche événementielle, c'est-à-dire qu'à chaque événement (par exemple, une ouverture de porte ou une variation dans la consommation électrique), sont recueillis l'information du changement d'état et le temps (à la seconde près) auquel l'événement est arrivé. Dans le réseau filaire du laboratoire DOMUS, le système est conçu pour recueillir l'état de tous les capteurs à intervalle fixe (par exemple à chaque seconde), qu'il y ait un changement d'état ou non.

Annotation et prétraitement des données recueillies.

M7. Prétraitement des données recueillies par le système d'acquisition de données lors des expérimentations;

M8. Annotation des données de capteurs en utilisant les données de l'application Android et les observations de l'expérimentateur;

Lors d'une expérimentation, une tablette tactile Android est ajoutée aux réseaux de capteurs avec laquelle l'expérimentateur peut noter le début et la fin de chacune des tâches. Pour ce faire, une application a été conçue qui permet de faciliter l'annotation des données. Cette application, nommée *Application Projet Dépistage (APD)*, affiche la liste de tâches auxquelles sont associées deux boutons chacune : « Démarrer » et « Arrêter ». L'application note le temps de début et de fin de l'expérimentation et chaque début et arrêt de tâche. L'interface d'APD est simple et facile à utiliser en situation de stress, car l'expérimentateur doit pouvoir rester concentré sur sa tâche première qui est d'observer le participant. L'expérimentateur prend aussi des notes manuscrites sur ses observations lors d'une expérimentation.

Les annotations d'APD sont utilisées de trois manières différentes. Tout d'abord, si on prend comme hypothèse que le temps requis pour accomplir une tâche par une personne est corrélé

avec la gravité de ses troubles cognitifs, il devient facile d'intégrer ce temps dans l'analyse des données. Ensuite, si on estime que seules certaines tâches sont discriminantes, il devient facile de découper les données pour isoler les capteurs qui ont été utilisés durant une tâche donnée. Cela permet par exemple de comparer l'exécution d'une tâche spécifique entre participants et de voir si des différences significatives sont observées. Finalement, grâce à APD, les temps exacts de début et de fin de chaque expérimentation sont disponibles, ce qui permet de filtrer les données superflues acquises par le système avant et après une expérimentation. En effet, il est impossible de démarrer l'acquisition des données exactement au moment où l'expérimentation commence, car un délai de démarrage du système est nécessaire.

Il est important de noter que toutes les expérimentations sont filmées. Au CRIUGM, l'expérimentateur filme les actions du participant à l'aide d'une caméra à l'épaule. Il peut se positionner le long d'un corridor d'où il peut filmer les actions du participant sans jamais être dans le champ de vision des capteurs de mouvement, ce qui est d'une importance capitale. Comme cette prémisses est impossible au laboratoire DOMUS, dû à la configuration des lieux, l'utilisation de caméras fixes est nécessaire. Puisque le laboratoire DOMUS possède déjà une caméra au plafond et un caméscope, cet équipement est complété par l'ajout de caméras IP contrôlables par Internet, fixées au mur. La technologie IP nous permet de démarrer l'acquisition de ces caméras de manière centralisée et au même moment. Par la suite, lorsque les expérimentations sont terminées, nous pouvons consulter les vidéos dans l'application web CoLab⁴. CoLab permet aux utilisateurs qui visionneront les vidéos d'ajouter des annotations aux vidéos sous la forme entre autres d'observations ponctuelles ou de verbatim. Ces observations sont typiquement catégorisées selon une grille d'analyse qui permet de les comptabiliser. Ces nouvelles informations, par exemple le nombre d'erreurs commises par le participant, peuvent venir compléter les données de capteurs utilisées pour le dépistage précoce de la DTA.

⁴ CoLab a été conçu par Jules Randolph pour le projet de dépistage précoce dans le cadre de sa maîtrise en informatique.

Analyse des données.

M9. Analyse des données traitées en tentant de prédire le diagnostic cognitif des participants à l'aide de techniques de classification automatique;

M10. Recherche de corrélations potentielles entre certaines variables observées et le diagnostic cognitif des participants à l'aide d'analyses statistiques.

L'approche préconisée pour l'analyse des données en vue du dépistage précoce de la DTA chez les participants est la classification automatique supervisée⁵. C'est une technique d'apprentissage simple qui nécessite l'étiquetage des données d'apprentissage selon leur classe. Dans notre cas, cette classe dépend du diagnostic cognitif officiel du participant d'une expérimentation donnée, soit : cognitivement sain, atteint d'un trouble cognitif léger (TCL) ou atteint d'une démence de type Alzheimer (DTA). Le but des algorithmes de classification utilisés est donc de prédire cette classe en analysant une partie des données qui n'est pas étiquetée. Pour ce faire, il faut, en premier lieu, trouver une façon de représenter les données pour qu'elles puissent être utilisées par l'algorithme. Les algorithmes de classification prennent en général un seul vecteur par instance d'une classe. Chacun constitue donc un exemple de données associées à la classe. Par contre, nos fichiers de données contiennent plutôt quelques milliers de lignes (vecteurs) par instance (expérimentation). Il faut alors trouver une façon de combiner des milliers de vecteurs en un seul. La solution pour la combinaison de ces vecteurs dépend de l'hypothèse sous-jacente à quelles difficultés cognitives seraient observables lors de la réalisation de tâches complexes. Par exemple, on peut vouloir s'appuyer sur une hypothèse disant qu'une personne atteinte d'un TCL ou d'une DTA aura plus de difficulté à se souvenir où vont les objets utilisés et ouvrira donc plus de portes d'armoires, par exemple, pour replacer ces objets que les personnes dites saines. Ainsi, les capteurs de contact sur les portes et les capteurs de mouvement devraient se déclencher significativement plus souvent. Dans ce cas, la solution utilisée sera de convertir les données en « nombre de déclenchements » de chaque

⁵ Bien entendu, l'apprentissage non supervisé pourrait aussi être envisagé, mais étant donné l'ampleur du projet dans son ensemble, nous avons préféré nous limiter à l'apprentissage supervisé dans le cadre de ma maîtrise.

capteur : chaque colonne représentant un capteur et chaque valeur du vecteur représentant le nombre de fois que ce capteur a changé d'état au cours de l'expérimentation. Cette hypothèse et une alternative, ainsi que leur traduction en algorithme d'apprentissage seront détaillées au chapitre 5. Enfin dans un premier temps, les algorithmes de classification utilisés seront de la famille des arbres de décision et des réseaux bayésiens.

Contributions

Comme le projet DEI repose sur une collaboration étroite entre plusieurs chercheurs, assistants de recherche et étudiants gradués, nous avons cru bon d'établir la liste de nos contributions spécifiques. Nous avons aussi identifié les objectifs que chaque contribution permet d'atteindre. Voici donc nos contributions:

- C1. Nous avons défini une méthode standardisée multisite pour mesurer et analyser les comportements de participants en commençant par l'élaboration d'un protocole rigoureux à suivre pour assurer la qualité des données acquises sur deux sites d'expérimentation. Ce protocole a été éprouvé par la tenue d'expérimentations et est reproductible sur d'autres sites.

- C2. Nous avons documenté et comparé les comportements des participants avec et sans troubles cognitifs lors de la réalisation d'AIVQ. Pour ce faire, nous avons conçu, implémenté et déployé un système d'acquisition de données hétérogènes pour récupérer les données des capteurs lors des expérimentations et une application sur tablette Android pour annoter les données en temps réel en y intégrant le temps requis par le participant pour accomplir chacune des tâches. Tout cela a été validé par la tenue d'expérimentations qui ont entre autres permis d'étudier la fiabilité des réseaux de capteurs et des données recueillies, ainsi que la synchronisation de sources de données hétérogènes.

- C3. Nos travaux d'analyse préliminaires indiquent qu'il serait possible de prédire le diagnostic cognitif des participants à partir des données recueillies par l'environnement intelligent à l'aide d'algorithmes de classification automatique. En effet, une approche de classification par arbre de décision simple a été utilisée sur les données d'expérimentation et montre une différence significative entre la classe des personnes saines et celle des personnes TCL en utilisant un ensemble réduit de capteurs discriminants. Par contre, aucune différence significative n'a été observée entre la classe des personnes DTA et les autres classes.
- C4. Nous avons déterminé une corrélation entre le temps requis pour compléter les tâches demandées et le diagnostic cognitif des participants. Certaines tâches, comme la préparation de repas, ont aussi révélé qu'une personne ayant des troubles cognitifs prendra plus de temps en moyenne pour réaliser la tâche. Ces résultats viennent des analyses statistiques faites par Maxime Lussier [12]. De plus, le nombre de déclenchements de capteurs serait aussi en lien avec le déclin cognitif des participants.
- C5. Nous avons vérifié la faisabilité de réaliser une expérimentation de cette envergure sur deux sites et la possibilité de générer des données similaires, comparables et pouvant être composées : les données des capteurs sans fil le sont. Pour ce faire, nous nous sommes assurés d'une certaine homogénéité de l'environnement et des données acquises entre les sites de Sherbrooke et de Montréal. Enfin, malgré des configurations spatiales différentes, les deux lieux sont comparables en termes de temps passé dans la pièce, ainsi qu'en utilisation des appareils de la cuisine et du rangement. Par contre, une différence significative est observée : les participants de Montréal passent plus de temps dans le salon qu'à Sherbrooke, toutes classes confondues. Divers aspects par rapport aux expérimentations multisites seront explorés par d'autres chercheurs et étudiants pour répondre à des questions comme « à quel point peut-on combiner les jeux de données de Sherbrooke et Montréal ? » et « peut-on utiliser les données de Sherbrooke pour l'apprentissage et celles de Montréal pour la validation de cet apprentissage ? ».

C6. Nous avons une idée du type optimal de réseaux de capteurs à utiliser et de la quantité optimale de capteurs pour obtenir des résultats significatifs. Les capteurs sans fil sont nettement moins fiables que les capteurs filaires, mais peuvent tout de même être utilisés. De plus, un petit nombre de capteurs bien choisis serait suffisant, dépendamment des analyses que l'on veut faire sur les données recueillies. Cependant, des analyses futures pourront vérifier si un nombre supérieur de capteurs peut aider à identifier des comportements inappropriés ou erratiques, comme ouvrir plusieurs fois une même armoire qui ne contiendrait rien de pertinent pour la tâche en cours.

Structure du mémoire

Ce mémoire est divisé en six chapitres. Dans le premier chapitre, nous présentons l'état de l'art sur les trois éléments au centre de nos travaux : le dépistage de la DTA, les habitats intelligents, le traitement et l'analyse de données issues de capteurs pour la classification. Il en ressortira qu'il n'existe pas actuellement de méthode de dépistage précoce de la DTA, qu'il semble qu'il existe des signes précurseurs au niveau cognitif, que ces signes seraient mis en exergue en situation de stress en milieu inconnu et que peut-être certaines tâches pourraient être discriminantes. En conséquence, le chapitre 2 présente le scénario d'expérimentation qui a été conçu pour explorer ces questions, ainsi que le protocole d'expérimentation qui a été mis en place pour s'assurer du bon fonctionnement des expérimentations étant donné la complexité des infrastructures spatiale, matérielle et logicielle impliquées. Ces infrastructures sont détaillées au chapitre 3 où sont présentés en particulier le système d'acquisition de données hétérogènes utilisé pour les expérimentations, le matériel utilisé et les différences entre les deux lieux d'expérimentation (DOMUS et CRIUGM). Recueillir les données brutes n'est pas suffisant, il faut pouvoir les annoter afin de mieux les valider, les comprendre et les analyser. C'est pourquoi le chapitre 4 traite spécifiquement de l'annotation des données issues des capteurs pour leur analyse. Ces annotations sont en particulier très utiles pour le prétraitement des données, qui est décrit au chapitre 5. Les données prétraitées sont finalement utilisées pour diverses analyses préliminaires présentées au chapitre 6. Ces analyses utilisent en particulier

l'apprentissage supervisé pour essayer de prédire la classe cognitive d'un participant. Enfin, nous terminons ce mémoire avec une conclusion qui rappelle nos contributions et qui esquisse plusieurs directions de recherche qu'il serait intéressant de poursuivre.

Chapitre 1

État de l'art

La revue de littérature présentée ici traite de ce qu'est la DTA, comment la dépister et montre deux projets réalisant des expérimentations au sein d'environnements intelligents, desquels nous nous sommes inspirés, tant du côté protocole d'expérimentation que du côté analyse de données.

1.1 La maladie d'Alzheimer

Selon la Société Alzheimer du Canada, la DTA est une maladie dégénérative qui se manifeste par une altération cognitive amenant des troubles liés à la mémoire, à la concentration, à la résolution des problèmes et au langage, suffisamment grave pour affecter la capacité de la personne atteinte à poursuivre ses activités quotidiennes [12], [13]. La DTA peut aussi avoir un impact sur les émotions et l'humeur de la personne atteinte, sur son comportement et sur ses capacités physiques, affectant la coordination et la mobilité de la personne [14]. Outre la DTA, il existe plusieurs autres maladies apparentées: comme les maladies vasculaires (dues à un accident vasculaire cérébral), la maladie à corps de Lewy et les traumatismes crâniens. Ces affections peuvent présenter des symptômes similaires ou qui se chevauchent.

Aux stades précoces, les personnes souffrant de la DTA présentent une atteinte plus spécifique de la mémoire à long terme des événements (mémoire épisodique) [13], suivi de près par une atteinte des fonctions exécutives [15], [16], [17], [18], [19], [20]. Les fonctions exécutives consistent en un ensemble de fonctions cognitives (ex. : inhibition, résolution de problèmes, flexibilité) qui permettent de réaliser et de contrôler des tâches nouvelles ou complexes [21]. À noter que la majorité de cette revue de littérature traitant de la DTA et de son dépistage a été menée par la chercheuse principale du projet, Nathalie Bier, et j'en reprends les points principaux.

1.1.1 Dépistage de la maladie d'Alzheimer

À ce jour, il n'existe pas de marqueurs biologiques qui permettent d'établir, sans équivoque, un lien entre leur présence chez des individus asymptomatiques et l'apparition subséquente de la maladie [3]. Deux grandes catégories de marqueurs biologiques sont généralement considérées dans le domaine médical [2]: la détection de faibles concentrations de peptides de type beta amyloïde (A β) dans le liquide céphalo-rachidien (obtenue via une ponction lombaire) et l'accumulation anormale d'un agent amyloïde observé lors d'un scan cérébral. Par contre, ces marqueurs manquent de sensibilité et des critères clairs de diagnostic basés sur ces marqueurs biologiques restent encore à être déterminés. Plusieurs limites ont aussi été soulevées dans les écrits à leur sujet. Une autre approche pour le dépistage consiste à utiliser des tests neuropsychologiques variés pour différencier les personnes ayant des troubles cognitifs de ceux qui n'en ont pas [3], [22]. Toutefois, ces tests cognitifs ne sont pas toujours en mesure de détecter des difficultés cognitives précoces et ont peu de pouvoir prédictif du fonctionnement réel de la personne dans sa vie quotidienne [3], [23].

Dans ce contexte, une évaluation fine et détaillée des AIVQ dans le cadre du vieillissement cognitif présente un potentiel certain pour contribuer au diagnostic précoce de la DTA. Cependant, plusieurs tests et questionnaires utilisés à ce jour ne permettent pas une compréhension détaillée des difficultés de la personne, sont biaisés par la propre perception de la personne et de ses proches, ou sont trop structurés pour permettre d'observer l'impact des troubles exécutifs sur le fonctionnement au quotidien. Une solution serait d'utiliser des capteurs et d'autres outils variés, au domicile ou en laboratoire, qui ont le potentiel de fournir des mesures novatrices et raffinées [7], [8], pouvant être acquises sur de longues périodes (plusieurs jours, voire plusieurs mois) et ce, sans que la personne soit dérangée dans ses activités [24]. Ces capteurs sont habituellement utilisés au sein de ce qu'on appelle des environnements intelligents, soit des environnements capables d'acquérir des données sur l'environnement et ses habitants, et d'exécuter des actions sur celui-ci.

Dans le contexte précis du dépistage précoce des troubles cognitifs, les initiatives se font rares. Les chercheurs d'une étude de 2008 [8] ont déployé, dans treize appartements de personnes

âgées, des détecteurs de vitesse de marche, un pilulier électronique ainsi qu'un ordinateur portable, dans le but de détecter des signes de troubles cognitifs sur une longue période de temps. La suite de leur étude suggère la faisabilité de déployer de tels dispositifs, de mesurer en continu et de recruter des personnes âgées intéressées par ces technologies [24]. Deux autres travaux sur le dépistage précoce des troubles cognitifs ont été réalisés plus récemment et sont à l'étude à la section 1.2 de ce mémoire.

Finalement, il est connu que les personnes âgées, particulièrement celles avec troubles cognitifs, éprouvent des difficultés particulières à s'adapter à la nouveauté, capacité dépendante des fonctions exécutives [25], [26]. Par exemple, les participants pourraient faire des déplacements inutiles ou des erreurs lors de la réalisation d'une tâche, qui se manifesteraient par des comportements relativement simples à observer par des indicateurs comme le temps de réalisation, l'ordre des étapes réalisées au sein d'une tâche, le nombre de déplacements, le nombre de portes ou d'armoires ouvertes et à quels moments, les étapes répétées d'une tâche, le temps d'exécution pour les sous-tâches, etc. Les technologies déployées au sein des sites d'expérimentations doivent être en mesure de reconnaître ces indicateurs.

1.2 Expérimentations au sein d'environnements intelligents

Il sera question dans cette section de deux travaux importants dans le cadre desquels se sont déroulées des expérimentations au sein d'environnements intelligents, dont nous nous sommes inspirés pour nos travaux.

1.2.1 Détection autonome et non intrusive d'un TCL chez les personnes âgées

Les travaux dirigés par Alex Mihailidis de l'*Institute of Biomaterials and Biomedical Engineering, University of Toronto* et publiés en 2015 [27] partent du principe précédemment découvert par Dodge et al. qu'il existe un lien entre le ralentissement de la vitesse de marche des personnes âgées et leur déclin cognitif [28]. Leur but était de discriminer les personnes

âgées saines des personnes atteintes d'un TCL en utilisant des informations comme la vitesse de marche, qui est inférée à partir des données de capteurs de mouvement à infrarouge. Ils ont obtenu de très bons résultats en utilisant une fenêtre de 24 semaines sur les données de 97 participants provenant de l'*ORegon Centre for Aging and TEChnology* (ORCATECH) [28]. Leurs travaux se démarquent, car ils utilisent des données du « vrai monde », ne provenant pas d'un laboratoire, mais plutôt de l'habitation des participants, ce qui est propice à mieux refléter la performance réelle des participants, vu qu'on les observe sur la durée et non qu'à un certain moment donné, de façon ponctuelle. Par contre, observer les participants sur une durée de 24 semaines s'avère fastidieux, car dès que l'on perd une semaine de données à cause de problèmes de capteurs par exemple, on doit tout recommencer.

Si l'on résume, dans ces travaux, les expérimentations se déroulent sur une longue période (trois ans) au sein de milieux connus des participants et chaque milieu peut être très différent des autres. De plus, ils n'utilisent qu'un seul type de capteur (mouvement à infrarouge). Contrairement à eux, nos travaux présentés dans ce mémoire visent plutôt à faire ressortir les difficultés dans la réalisation d'AIVQ en réalisant des expérimentations en milieu inconnu et sous le stress. Ces expérimentations seraient de courte durée (environ une heure) et se dérouleraient en milieu standardisé en demandant aux participants d'exécuter une liste standardisée d'AIVQ discriminantes.

1.2.2 Évaluation autonome de la santé cognitive en utilisant la domotique

Les travaux du centre de recherche de Diane J. Cook, au *School of Electrical Engineering and Computer Sciences* du *Washington State University*, publiés en 2013 [29], ont comme hypothèse de départ : un déclin cognitif peut être mis en évidence à travers l'observation de la performance dans la réalisation d'AIVQ. Les objectifs de ces travaux sont de discriminer les personnes âgées saines des personnes atteintes d'un TCL ou d'une DTA, et d'évaluer la performance de réalisation des AIVQ. À la suite de leurs analyses, ils parviennent à différencier les personnes atteintes d'une DTA des personnes saines, mais leurs algorithmes ne trouvent pas de différence significative avec les personnes atteintes d'un TCL.

En résumé, ces travaux ressemblent à ce que l'on veut faire : inviter des personnes âgées cognitivement saines, des personnes atteintes d'un TCL et d'autres d'une DTA, à venir réaliser des AIVQ dans un laboratoire intelligent, pour observer leurs comportements et difficultés à l'aide de capteurs placés dans l'environnement. Par contre, quelques points les distinguent de nos idées :

- Les activités demandées sont exécutées une à la suite de l'autre, séquentiellement. Donc, l'expérimentation est arrêtée puis redémarrée pour chaque tâche.
- Les participants font la visite de l'appartement avant de commencer. Donc, l'environnement est alors déjà connu des participants lors de la réalisation des tâches.
- Les expérimentations se déroulent dans un seul laboratoire, celui du CASAS au *Washington State University*. Donc, pas d'étude de faisabilité de réalisation d'expérimentations multisites.

Dans notre protocole, les AIVQ demandées aux participants pourront être faites en parallèle, dans l'ordre désiré. De plus, les participants ne peuvent pas voir à quoi ressemble l'environnement d'expérimentation avant le début de l'expérimentation. Il s'agit donc de mettre à l'épreuve les capacités de planification, de gestion du temps et d'adaptation à la nouveauté des participants. Finalement, notre protocole d'expérimentation est standardisé pour être utilisé sur plusieurs sites d'expérimentation : aux laboratoires DOMUS et du CRIUGM.

1.3 Conclusion

Nous pouvons conclure en rappelant qu'il n'existe pas encore de moyen fiable et pratique pour le dépistage précoce de la DTA et qu'il semble exister des signes précurseurs d'un déclin cognitif qui seraient mis en évidence lors de la réalisation d'AIVQ discriminantes au sein d'un environnement intelligent inconnu des participants. Bien que chaque projet dans le domaine se dote d'une méthodologie de recherche appropriée et systématique, basée évidemment sur la méthode scientifique, à notre connaissance personne n'a proposé jusqu'à maintenant une

méthodologie standardisée multisite pour mesurer et analyser les comportements de participants. Nos travaux diffèrent aussi au niveau de la durée d'une expérimentation, de la connaissance a priori des lieux, de la séquentialité (ou non) des AIVQ à réaliser et de la naïveté des capteurs utilisés. C'est précisément ce vers quoi les travaux présentés dans ce mémoire se dirigent.

Chapitre 2

Protocole d'expérimentation

Pour tester notre idée d'utiliser un environnement intelligent pour le dépistage précoce de la DTA, la réalisation d'expérimentations est nécessaire. Commençons par détailler le déroulement de ces expérimentations. Nous verrons ensuite les conditions initiales communes pour chaque expérimentation et le protocole mis en place pour les respecter. Une attention particulière a été portée à unifier le protocole d'expérimentation pour les deux laboratoires impliqués dans le projet et le rendre reproductible pour de futures recherches. Sauf si c'est spécifié autrement, toutes les démarches détaillées ici s'appliquent aux deux laboratoires et pourraient éventuellement s'appliquer à d'autres.

2.1 Scénario d'expérimentation

Une expérimentation comprend deux parties : des tests cognitifs et la réalisation d'une liste de tâches dans l'appartement intelligent. Les tests cognitifs et la liste de tâches sont détaillés ici, ainsi que de leur déroulement.

2.1.1 Tests cognitifs

Dans le cadre de leur participation à notre projet, les participants sont sujets à des tests cognitifs précédant l'expérimentation en laboratoire (Tableau 1 et Tableau 2) servant à confirmer leur diagnostic cognitif. Nous vérifions donc si les participants n'ont pas subi de déclin cognitif depuis l'émission de leur plus récent diagnostic. Ces tests sont administrés par une ergothérapeute et sont d'une durée variant entre une et deux heures, les participants ayant plus de difficultés prenant habituellement plus de temps. Il s'agit de tests standardisés, ainsi l'administration de chaque test est uniforme. La plupart des tests se font sur place, sans support informatique. Par contre, les tests « mémoire de la source » et « test de dénomination » se font

sur ordinateur, et les tests « CRIq » et « Chute - falls efficacy scale » peuvent se faire au téléphone pour sauver du temps, si nécessaire. De plus, les participants atteints de TCL doivent passer trois tests supplémentaires et répondre à deux questions supplémentaires (Tableau 2).

Tableau 1 – Tests cognitifs standardisés administrés à tous les participants

Test	Méthode de passation
MoCA	Sur place
Geriatric depression scale	Sur place
CRIq	Par entrevue téléphonique ou sur place
trail-making test A et B	Sur place
Index de qualité du sommeil (IQSP 1.0)	Sur place
Histoire oculaire	Sur place
Chute - falls efficacy scale	Par entrevue téléphonique ou sur place
15 mots de Rey (histoire A)	Sur place
D-KEFS - stroop	Sur place
D-KEFS - tour de Londres	Sur place
Mémoire de la source	Sur place, par ordinateur
Anxiété	Sur place
Test de dénomination	Sur place, par ordinateur

Tableau 2 – Tests et questions supplémentaires administrés aux participants TCL seulement

Tests supplémentaires	Méthode de passation
Boston naming test	Sur place
BORB - OD A hard	Sur place
Fluence verbale TNP-animaux	Sur place
Questions supplémentaires	Méthode de passation
« Pensez-vous avoir des problèmes de mémoire ? »	Sur place, à l'oral
« Est-ce que cela affecte votre quotidien ? »	Sur place, à l'oral

Généralement, les tests sont administrés le jour même de l'expérimentation en laboratoire, les tests précédant l'expérimentation. Si le participant se sent trop fatigué à la suite des tests, il pourra continuer une prochaine fois et sera alors rencontré de nouveau la semaine suivante. Il commencera la visite directement par l'expérimentation en laboratoire.

Il est important de noter que la quantité de tests administrés avant l'expérimentation en laboratoire est considérable et peut entraîner de la fatigue chez le participant. Ses performances en laboratoire peuvent en être affectées. Un participant qui effectuera les tests cognitifs et les tâches en laboratoire en deux séances distinctes pourrait donc mieux réussir les tâches demandées.

2.1.2 Tâches observées

Les expérimentations se déroulant dans le cadre de ce projet consistent en l'observation d'une personne âgée pendant qu'elle accomplit une liste non ordonnée de tâches bien précises. Les tâches demandées ont été réfléchies et choisies en se basant sur les études sur le Profil des AVQ [6] et sur le Profil des Activités Instrumentales [30] et devraient mettre en évidence les difficultés cognitives des participants, car chacune sollicite fortement certaines fonctions exécutives. Par exemple, la tâche « préparer un repas » requiert beaucoup de planification et une bonne gestion des risques, étant donné l'utilisation de la cuisinière. De plus, la tâche « répondre au téléphone » interrompt le participant dans son fil d'activités et lui demande de replanifier la suite selon la nouvelle tâche demandée. Pour faciliter l'analyse des données issues des capteurs, les tâches doivent pouvoir se réaliser dans des endroits différents de l'appartement. Qui plus est, certaines de ces tâches ont également déjà été testées en laboratoire par certains membres de notre équipe.

Ainsi, les participants disposent de 45 minutes pour réaliser, dans l'ordre qui leur convient, toutes les activités suivantes:

- Déposer ses effets personnels (manteau et sac à main) dans la garde-robe;
- Préparer un repas léger, plus précisément, il s'agit de
 - Préparer un œuf, deux toasts avec de la confiture pour 1 personne;
 - Préparer 1 tasse de café avec lait et sucre;
 - Dresser la table;
 - Ranger et nettoyer après le repas;
- Nettoyer la salle de bain, plus précisément, il s'agit de
 - Tirer la chasse d'eau de la toilette;
 - Laver le miroir et le lavabo avec un produit nettoyant;
- Téléphoner pour obtenir l'horaire des départs d'autobus pour Montréal⁷ durant la journée et le noter sur une feuille;
- Répondre au téléphone qui sonnera pendant l'expérimentation et exécuter la tâche indiquée par l'interlocuteur.

2.2 Déroulement des expérimentations

Nous exposerons ici notre protocole pour la tenue d'expérimentations en environnement intelligent où l'on observe le comportement des participants. Il sera question des conditions initiales à respecter avant la tenue d'une expérimentation, du déroulement des expérimentations et des vérifications à faire ensuite.

2.2.1 Conditions initiales d'expérimentation

En respect de la méthode scientifique, et afin d'isoler et de contrôler les paramètres d'expérimentation, les conditions initiales d'expérimentation sont l'ensemble des prérequis pour garantir que, d'une expérimentation à l'autre :

⁷ Toronto si l'expérimentation a lieu à Montréal.

- les participants évoluent dans un environnement qui reste toujours le même d'un participant à l'autre ;
- les données sont récoltées à partir de capteurs dont le positionnement et la configuration demeurent constants.

On distinguera les conditions initiales techniques, qui sont relatives au système d'acquisition de données, et les conditions initiales environnementales, relatives à l'état général de l'appartement, la disposition des objets et la propreté.

Comme l'un des objectifs du projet est de vérifier la faisabilité de réaliser une expérimentation sur deux laboratoires intelligents, il a fallu mettre en œuvre des dispositions qui tendent vers l'uniformisation des deux laboratoires. Il était bien entendu impossible de modifier la disposition des pièces de chacun des appartements, mais le choix de l'emplacement de certains appareils et plusieurs objets était de notre ressort⁸. Cela a été fait pour limiter les biais causés par les différences entre les deux environnements, car nous voulons vérifier la possibilité de générer des données similaires, comparables et agrégeables.

Ainsi, nous avons élaboré une liste de tâches à faire avant chaque expérimentation pour s'assurer que les conditions initiales sont réalisées (Annexe B.1). Elle se subdivise en deux grandes catégories : conditions initiales environnementales et considérations techniques. La première catégorie est divisée selon les pièces de l'appartement et indique quels objets importants doivent être à quels endroits pour s'assurer que les participants soient tous confrontés aux mêmes difficultés. La seconde catégorie représente les vérifications à faire du côté technologique, comme vérifier le fonctionnement des capteurs et des applications à démarrer juste avant de commencer l'expérimentation. On rappelle aussi qu'on doit au préalable recharger certains appareils à pile, comme la tablette ou les caméscopes, plusieurs heures avant usage.

⁸ Dans la cuisine, les assiettes, les tasses, le café, le sucre, etc. ont été placés à l'intérieur d'armoires à proximité de la cuisinière et suivant une logique semblable pour les deux laboratoires. Par exemple, les assiettes ont été placées dans la première armoire en haut à droite de la cuisinière, si on fait face à celle-ci.

La liste de vérifications préexpérimentation peut être sujette à modification au fil du temps. Par exemple, lors d'une expérimentation, le détecteur de fumée de la cuisine s'est déclenché prématurément. Cela a causé un grand stress au participant qui ne savait pas quoi faire pour que le détecteur de fumée arrête de sonner. L'expérimentateur a dû venir et l'arrêter, causant de faux déclenchements de capteurs de mouvement. C'est pourquoi nous avons dès lors décidé de toujours désactiver le détecteur de fumée avant chaque expérimentation.

2.2.2 Lors d'une expérimentation

Le jour venu, lorsqu'un participant arrive, l'expérimentateur l'accueille et le participant fait connaissance avec l'équipe technique. Nous prenons soin de ne pas permettre au participant de visiter, ni même voir l'habitat intelligent avant de débiter l'expérimentation pour que l'environnement soit tout à fait nouveau et inconnu au début de l'expérimentation. Ensuite, l'expérimentateur enchaîne avec les tests cognitifs décrits à la section 2.1.1.

Puis, juste avant de débiter l'expérimentation, le participant et l'expérimentateur se retrouvent devant la porte d'entrée de l'habitat intelligent et la liste de tâche est présentée au participant. Celui-ci a quelques minutes pour se familiariser avec la liste et poser des questions sur les tâches à faire. Lorsqu'il est prêt, le participant en avertit l'expérimentateur et ouvre la porte du laboratoire. Cela marque le début de l'expérimentation. L'expérimentateur marque l'heure de début à l'aide de la tablette Android. À noter qu'au laboratoire de Montréal, la tablette Android n'est pas utilisée, donc l'heure de début est notée sur papier.

Durant l'expérimentation, l'expérimentateur n'intervient en aucun cas, sauf si le participant met en danger sa sécurité. En tout temps, l'expérimentateur peut entrer en contact avec la sécurité en cas d'urgence ou de danger. Il peut intervenir également si la personne est dans une impasse ou si la personne lui demande de l'aide. Si le participant n'a pas terminé après quarante-cinq minutes, un délai de quinze minutes supplémentaires lui est accordé. De l'aide lui est ensuite graduellement apportée s'il n'a toujours pas terminé après le délai supplémentaire. Si la personne éprouve de grandes difficultés ou qu'une détresse est observée,

un temps d'arrêt est suggéré avec soutien de la part de l'expérimentateur. La tâche pourra être reprise ou arrêtée selon le cas.

Quelques considérations supplémentaires sont prises lors des expérimentations au DOMUS pour s'assurer du déroulement identique des expérimentations. Par exemple, au moment où le participant commence à faire cuire un œuf sur la cuisinière, l'expérimentateur avertit son collègue à l'extérieur (par message texte) pour qu'il interrompe le participant en appelant sur le téléphone du salon. L'appel doit arriver à ce moment précis, car nous voulons tester si le participant saura prendre une décision sécuritaire (comme éteindre le rond avant d'aller répondre au téléphone). Dans le cas contraire, il pourrait s'agir d'un manque d'attention ou une mauvaise gestion du risque plus commune chez les personnes atteintes de troubles cognitifs.

À la fin de l'expérimentation, lorsque le participant affirme avoir terminé, l'expérimentateur appuie sur le bouton « Terminer le scénario » de la tablette Android. Cela marquera l'heure de fin de l'expérimentation. Ensuite, l'expérimentateur invite le participant à s'asseoir quelques instants pour discuter de ses impressions sur les tâches accomplies et sur sa performance, puis le participant sera raccompagné.

2.2.3 À la suite d'une expérimentation

Lorsque l'expérimentation est terminée, l'équipe technique s'affaire en premier lieu à faire un tour rapide de l'appartement pour vérifier si tout a été bien rangé à la bonne place et de façon sécuritaire (appareils éteints, etc.). S'en suit une série de tâches à faire pour bien clore chaque expérimentation. À l'instar de la liste préexpérimentation, nous avons rédigé une liste de tâches à faire après chaque expérimentation pour assurer que l'acquisition de données soit arrêtée et que toutes les données soient bien stockées (Annexe B.2). Elle se divise en deux catégories : les vérifications à faire dans l'appartement (comme arrêter le caméscope et récupérer la vidéo) et les manipulations à faire sur le système d'acquisition. Une fois l'acquisition arrêtée et toutes les données stockées et vérifiées, il ne manque plus qu'à revérifier la liste préexpérimentation pour continuer avec un nouveau participant.

2.3 Automatisation des vérifications technologiques

Comme on a pu le constater, une multitude de tâches doivent être faites pour s'assurer que tout est en place pour la tenue d'une expérimentation. Lorsque celle-ci est terminée, plusieurs autres tâches doivent être accomplies pour s'assurer que les données sont bien sauvegardées. Rappelons que ces tâches sont divisées en deux sections : les conditions initiales de l'expérimentation et les considérations techniques. La section de la liste qui concerne les conditions initiales de l'expérimentation est, quant à elle, obligatoire. Sa systématisation garantit, d'une part, l'homogénéité des expérimentations sur un même site; d'autre part, elle rend comparables entre elles les conditions initiales entre chaque site. De plus, elle devrait faire partie intégrante du protocole de recherche, et sa description devrait se faire lors d'expérimentations de test.

Malgré la mise en place de listes à cocher claires et complétées avec assiduité, des erreurs humaines peuvent survenir vu la quantité de tâches répétitives à effectuer. De plus, l'accomplissement des tâches de ces listes requiert un temps considérable avant et après chaque expérimentation. C'est pour cela que la section de la liste qui concerne les considérations technologiques devrait être automatisée. Pour ce faire, il serait souhaitable d'utiliser une interface de programmation applicative [31] (en anglais *Application Programming Interface*, ou API) qu'on appellera « API Scénarii ». Celle-ci serait générique, donc utilisable avec divers habitats intelligents et non couplée à un environnement d'acquisition donné. Cette API définit des caractéristiques obligatoires à respecter par les modules d'acquisition de tous les sites où se déroulent des expérimentations dans un même projet, telle que la synchronicité entre les modules (ex.: vidéos, capteurs, activités...). Cette proposition fait l'objet d'une réflexion plus approfondie dans le mémoire de mon collègue Jules Randolph de l'UdeS.

2.4 Comparaison des sites

Ce document a tenté jusqu'à maintenant d'expliquer le protocole d'expérimentation de façon le plus générique possible. Par contre, quelques différences ont été notées entre les deux laboratoires. Vous verrez ces différences dans le Tableau 3 suivant.

Tableau 3 – Différences entre les laboratoires DOMUS et du CRIUGM

DOMUS (Sherbrooke)	CRIUGM (Montréal)
Un assistant de l'expérimentateur annote les données pendant l'expérimentation à l'aide d'une tablette Android	Pas d'annotation des données
Tâche du téléphone : demander horaire d'autobus pour Montréal	Tâche du téléphone : demander horaire d'autobus pour Toronto
Un assistant de l'expérimentateur appelle le participant pendant l'expérimentation sur le téléphone du salon au moment précis où celui-ci exécute la tâche de la cuisson de l'œuf. L'objectif est de l'interrompre durant une tâche potentiellement dangereuse.	L'expérimentateur étant seul, il effectue l'appel dès qu'il a l'occasion de s'éloigner pour faire l'appel lui-même sans rater un moment important. L'appel n'interrompt donc pas nécessairement une tâche potentiellement dangereuse.
Six caméras fixes pour filmer l'expérimentation (chambre, salle de bain, salle à manger et trois dans la cuisine)	Une caméra à l'épaule de l'expérimentateur
Deux réseaux de capteurs : filaires et sans fil (voir Chapitre 3)	Réseau de capteurs sans fil uniquement (voir Chapitre 3)

2.5 Conclusion

Nous avons vu dans ce chapitre le protocole et le déroulement des expérimentations, avec l'accent mis sur le contrôle des conditions initiales d'expérimentation et l'aspect multisite. Nous avons aussi explicité des différences au niveau du protocole d'expérimentation entre les deux laboratoires, principalement causées par l'environnement d'expérimentation. Il est

maintenant temps d'enchaîner avec la description des environnements intelligents utilisés pour le déroulement de ces expérimentations et des technologies utilisées.

Chapitre 3

Environnement et architecture

Une fois le protocole d'expérimentation établi en fonction des environnements intelligents à notre disposition, il faudra concevoir, implémenter et déployer le système d'acquisition de données pour recueillir des mesures objectives sur la réalisation des AIVQ lors des expérimentations. Ce chapitre commencera par détailler en quoi consistent les environnements intelligents des laboratoires DOMUS et du CRIUGM, de quel matériel ils sont équipés et comment les pièces sont disposées. Les différences entre les deux lieux d'expérimentations seront explicitées. Nous verrons ensuite l'architecture du système d'acquisition de données hétérogènes, qui se divise en un réseau de capteurs filaires précédemment installé au laboratoire DOMUS et un réseau de capteurs sans fil déployé dans le cadre de ce projet.

3.1 Environnement

L'environnement intelligent du laboratoire DOMUS est un appartement de style 3½ meublé (Figure 2), situé sur le campus de l'Université de Sherbrooke. Cet appartement a été transformé en un environnement intelligent en deux étapes. Tout d'abord, il a été équipé de réseaux hétérogènes (Ethernet, WiFi, Bluetooth, Z-Wave, courants porteurs, infrarouges, etc.), capteurs et effecteurs (détecteurs de mouvements, contacts électromagnétiques, tapis tactiles, débitmètres, microphones, haut-parleurs, lecteurs et étiquettes RFID, etc.), appareils de cuisine (cuisinière, lave-vaisselle, micro-ondes, grille-pain, etc.), mobilier d'appartement (table, chaises, armoires, lit, etc.) et objets de communication (écrans tactiles, PDA, téléphones cellulaires, etc.). La salle des serveurs, les murs et le plafond de l'appartement ont été conçus pour faciliter la réorganisation et la reconfiguration des capteurs et des effecteurs. Deuxièmement, des systèmes d'assistance cognitive, de surveillance à distance et médicaux ont été utilisés [32].

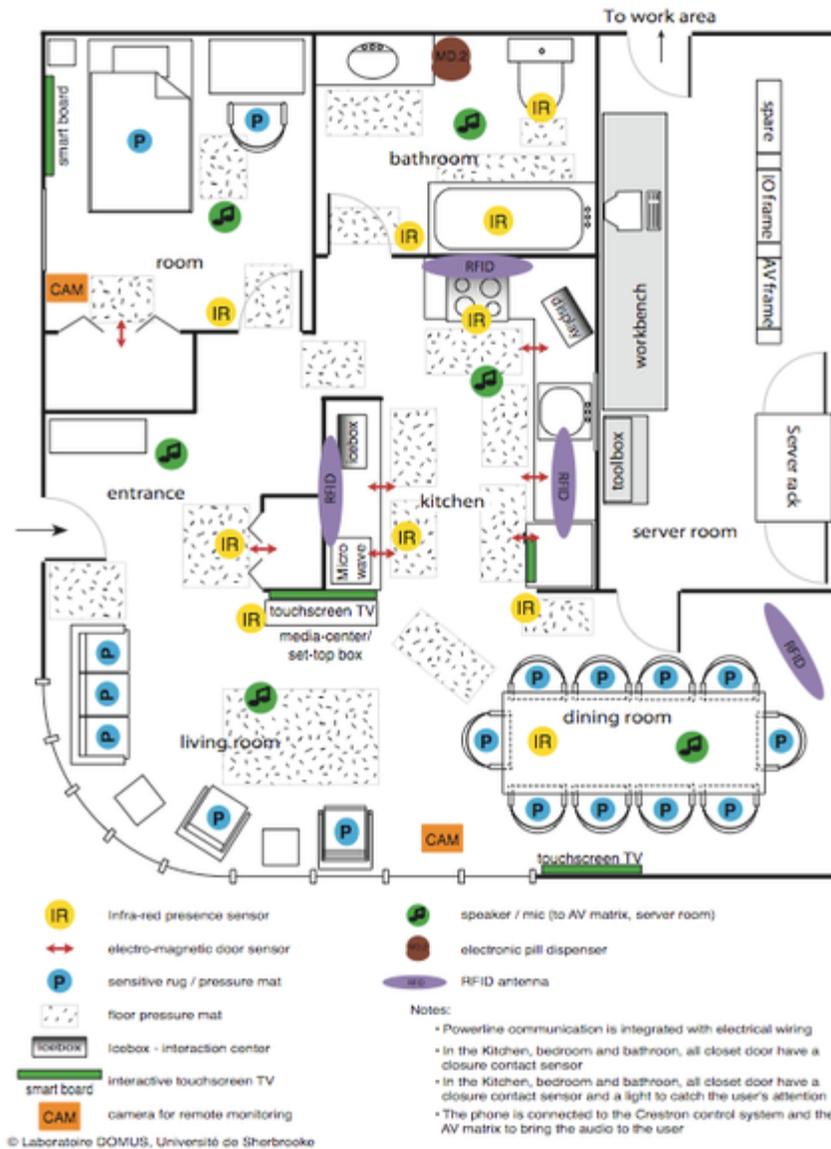


Figure 2 – Plan du laboratoire DOMUS

L'environnement intelligent du CRIUGM est aussi un appartement de style 3½ meublé (Figure 3), mais moins équipé. L'appartement est fait sur la longueur, avec un corridor. La cuisine, la salle à manger et le salon sont séparés par des paravents. La salle de bain et la chambre sont séparées par un demi-mur. Cet appartement est aussi utilisé pour des projets en ergothérapie et en physiothérapie, donc la séparation des pièces a été pensée en conséquence.

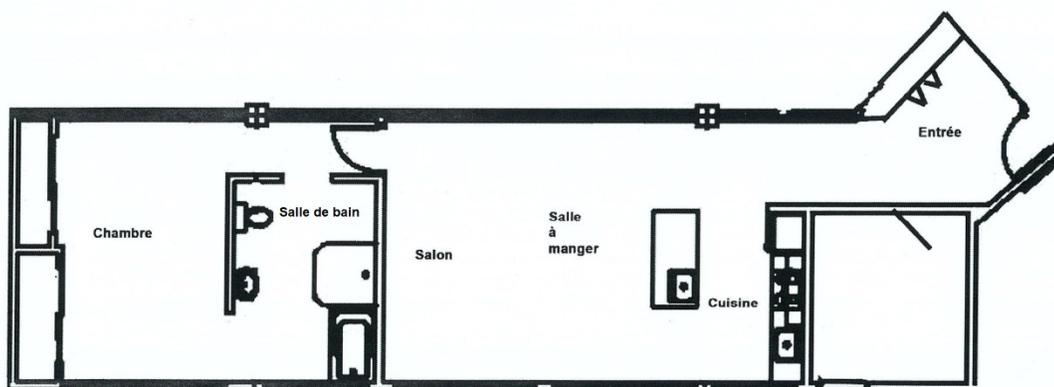


Figure 3 – Plan du laboratoire du CRIUGM

Comme on peut le constater, les deux appartements sont conçus très différemment. Nous détaillerons les différences au niveau des réseaux de capteurs et de l'acquisition vidéo et audio dans les prochaines sections.

3.1.1 Réseau de capteurs filaires

L'appartement intelligent du laboratoire DOMUS a été conçu il y a plusieurs années pour servir d'environnement d'expérimentation pour une multitude de projets. De ce fait, il a été équipé d'une foule de capteurs, d'appareils et de gadgets utiles à des fins d'observation des occupants ou d'interaction avec ceux-ci. Dans le cadre de ce projet, nous avons utilisé l'infrastructure système du laboratoire telle quelle. Précisons que toutes les portes d'armoire et les tiroirs sont équipés d'un capteur de contact électromagnétique, toutes les salles sont surveillées par des capteurs de mouvement, tous les robinets ont des débitmètres, etc. On peut voir la disposition des pièces et le placement des capteurs de mouvement (entre autres) dans le plan officiel du laboratoire (Figure 4). Pour compléter l'information présentée, le plan des élévations du laboratoire (Figure 5) offre une vue de face des différentes armoires et tiroirs de l'appartement ainsi que les capteurs présents.

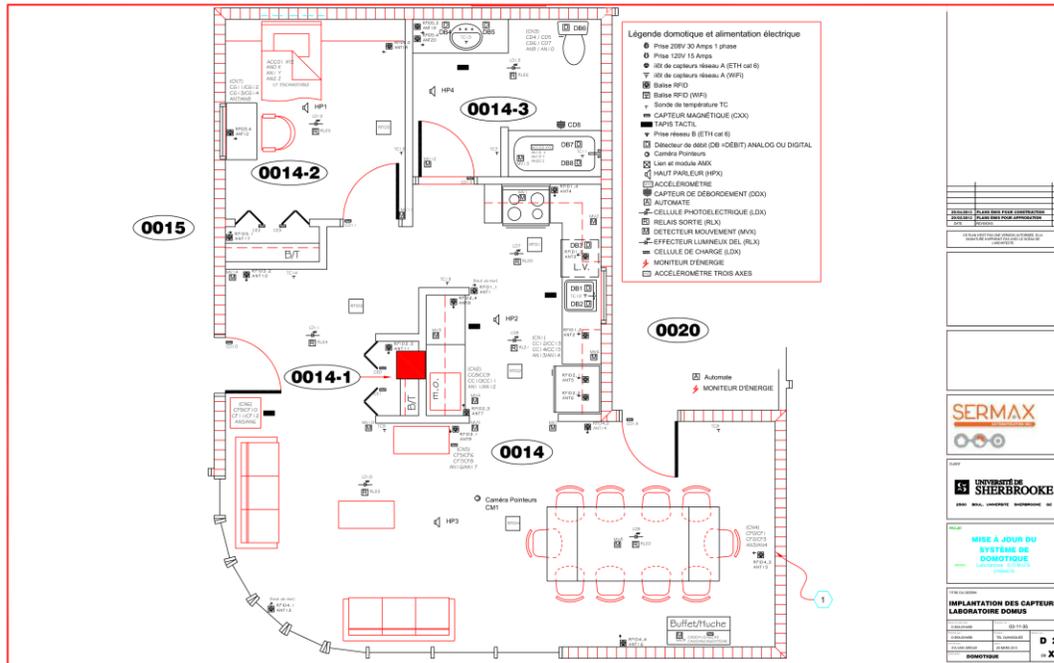


Figure 4 – Plan de sol officiel du laboratoire DOMUS

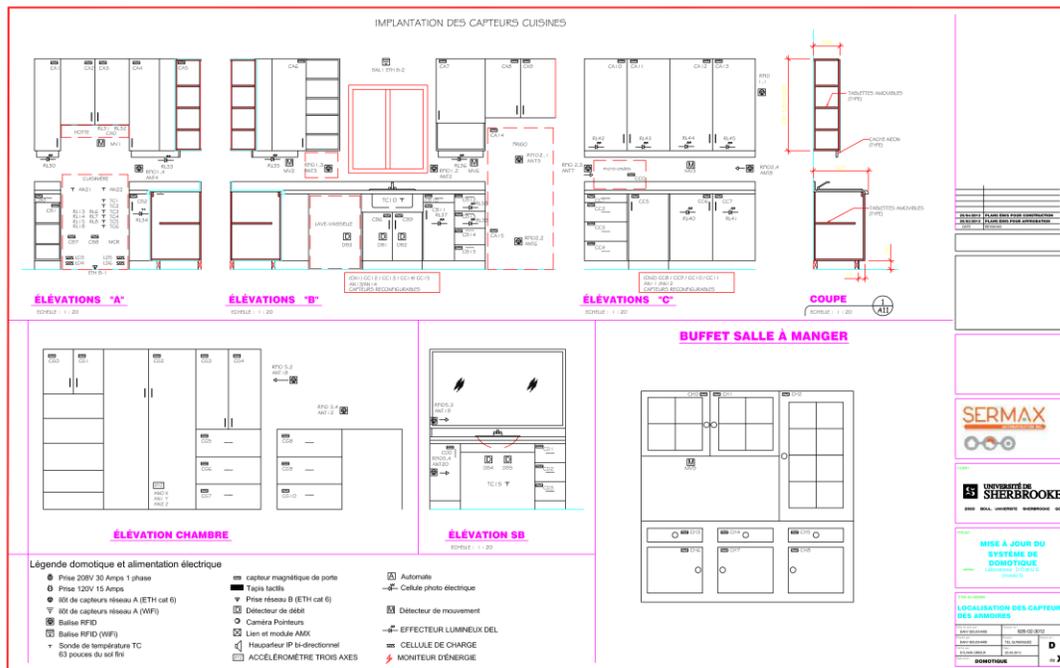


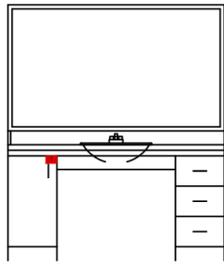
Figure 5 – Élévations des armoires et meubles de l'appartement du laboratoire DOMUS

Il est bon de rappeler qu'aucun de ces capteurs filaires n'est présent au laboratoire du CRIUGM. L'utilisation du réseau filaire dans ce projet-ci est donc pour s'assurer d'avoir des données de capteurs de qualité, issues d'un réseau de capteurs fiable et éprouvé, et de ne laisser passer aucune information qui pourrait être utile.

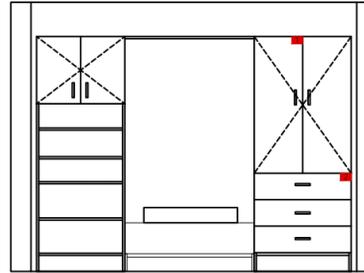
3.1.2 Réseau de capteurs sans fil

En plus du réseau de capteurs filaires, un réseau de capteurs sans fil est ajouté au laboratoire DOMUS. Un réseau similaire, constitué des mêmes modèles et types de capteurs, est installé au laboratoire du CRIUGM. Le réseau de capteurs sans fil est moins exhaustif que le réseau filaire, composé que de capteurs de contact électromagnétique, de mouvement et de consommation électrique, le but étant de rester comparable avec la faible quantité de capteurs utilisée au CRIUGM, de déterminer quels capteurs sont essentiels et de faciliter l'installation d'un tel réseau pour de projets futurs.

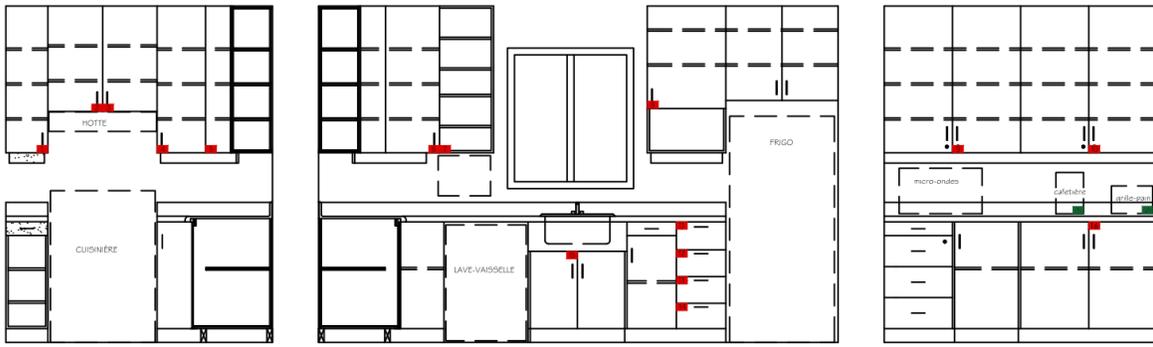
Pour commencer, les capteurs de contact électromagnétiques (Figure 6) sont placés sur la plupart des portes d'armoires susceptibles d'être ouvertes. Nous avons à notre disposition un nombre limité de capteurs, donc il était impossible d'en poser sur toutes les portes. Dans la salle de bain, seule la porte d'armoire dispose d'un capteur, car le participant n'a besoin de trouver comme objet que du produit nettoyant, et cela se trouve rarement à l'intérieur de tiroirs. Pour la chambre, un capteur est placé sur le premier tiroir de la commode de droite en partant du haut (celui où le participant devra ranger les vêtements) et sur une des portes d'armoire au-dessus. Dans la cuisine, presque toutes les portes et tiroirs ont été équipées d'un capteur. Pour les armoires larges ayant deux portes, un seul capteur est mis systématiquement sur une des deux portes, car la probabilité que le participant ouvre les deux portes en même temps ou séquentiellement dans un court laps de temps est élevée. Les autres armoires et tiroirs n'ont pas de capteurs, car ils sont difficiles à atteindre ou ne contiennent rien d'intéressant pour le participant. Pour les capteurs de consommation électrique, ceux-ci sont placés sur la cafetière et le grille-pain. Nous n'avons pas mis de capteur sur le four à micro-ondes, car les participants n'en avaient pas besoin et il n'y en a pas dans le laboratoire du CRIUGM.



Salle de bain



Chambre



Cuisine

Légende	
capteur de contact	■
détecteur d'électricité	■
Echelle en mm	1000

Figure 6 – Emplacement des capteurs de contact et de consommation électrique

Pour ce qui est des capteurs de mouvement (Figure 7), la règle générale était d'avoir un capteur qui couvre chaque salle. La raison est simple : si une tâche ne se déroule normalement que dans une pièce, il sera possible de déduire que le participant commence à exécuter cette tâche dès que le capteur de mouvement approprié se déclenche. Pour le laboratoire DOMUS, la plupart des capteurs de mouvement sans fil ont été placés juste à côté des capteurs filaires, sur le mur près du plafond, car ceux-ci étaient déjà placés à des endroits optimaux. Pour les capteurs du salon et de la salle à manger, ceux-ci ont été placés sur le plafond, au-dessus du divan et de la

table de cuisine respectivement, afin de laisser un corridor (voir l'encadré en vert dans la Figure 7) pour que l'expérimentateur puisse se déplacer sans être vu par les capteurs de mouvement sans fil. En effet, un déclenchement de capteur de mouvement causé par l'expérimentateur viendrait fausser les données, nous faisant croire que le participant est rendu dans le salon ou dans la salle à manger sans qu'il n'y soit vraiment.

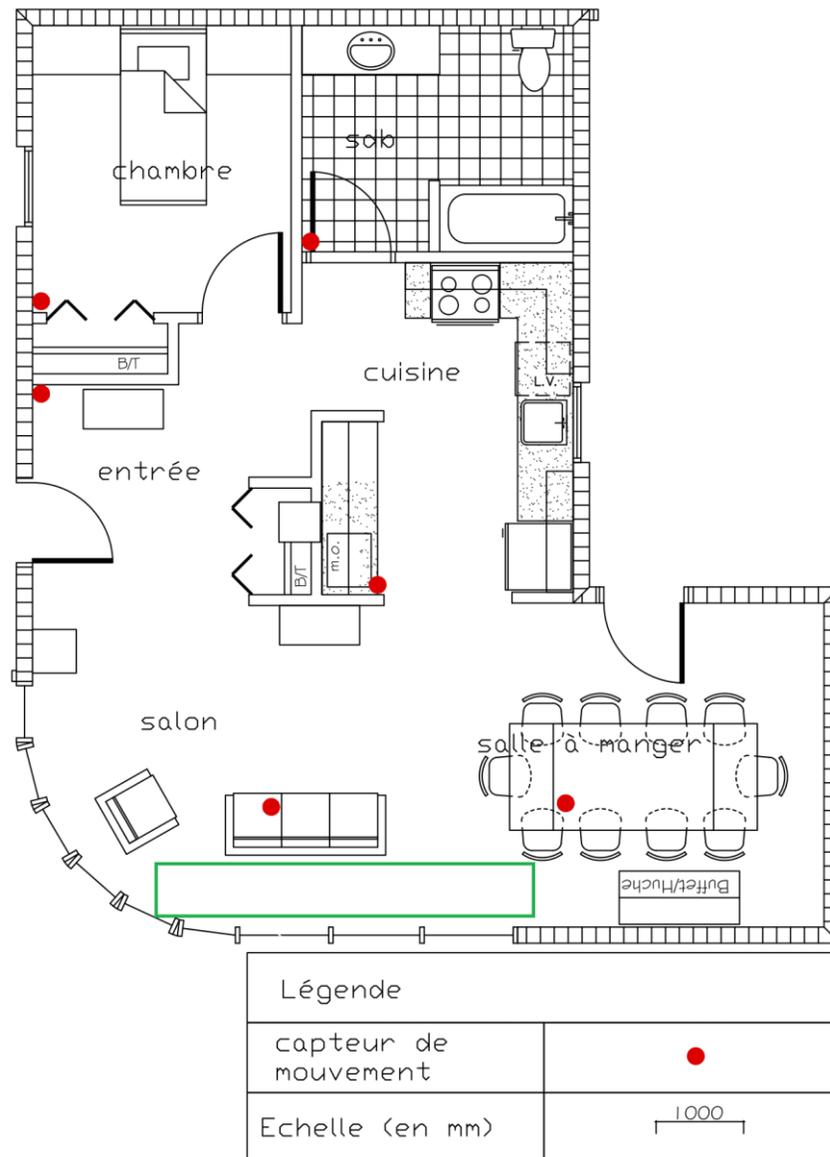


Figure 7 – Emplacement des capteurs de mouvement sans fil

Dans l'appartement du CRIUGM, le même concept de corridor pour l'expérimentateur a été pensé (voir l'encadré en vert dans la Figure 8). Étant donné la disposition des pièces, cela n'a nécessité qu'un travail minime. Un capteur de mouvement a aussi été placé par pièce. Le salon, la salle à manger et la cuisine sont séparés par des paravents; les capteurs de mouvement sont installés sur ces paravents. Pour ce qui est des capteurs de contact, ils sont placés sur le frigo, les portes d'armoire à gauche et à droite de la cuisinière, ainsi que sur le tiroir de la commode de la chambre. Les capteurs de consommation électrique sont branchés sur le grille-pain et la cafetière, sur l'îlot de la cuisine.

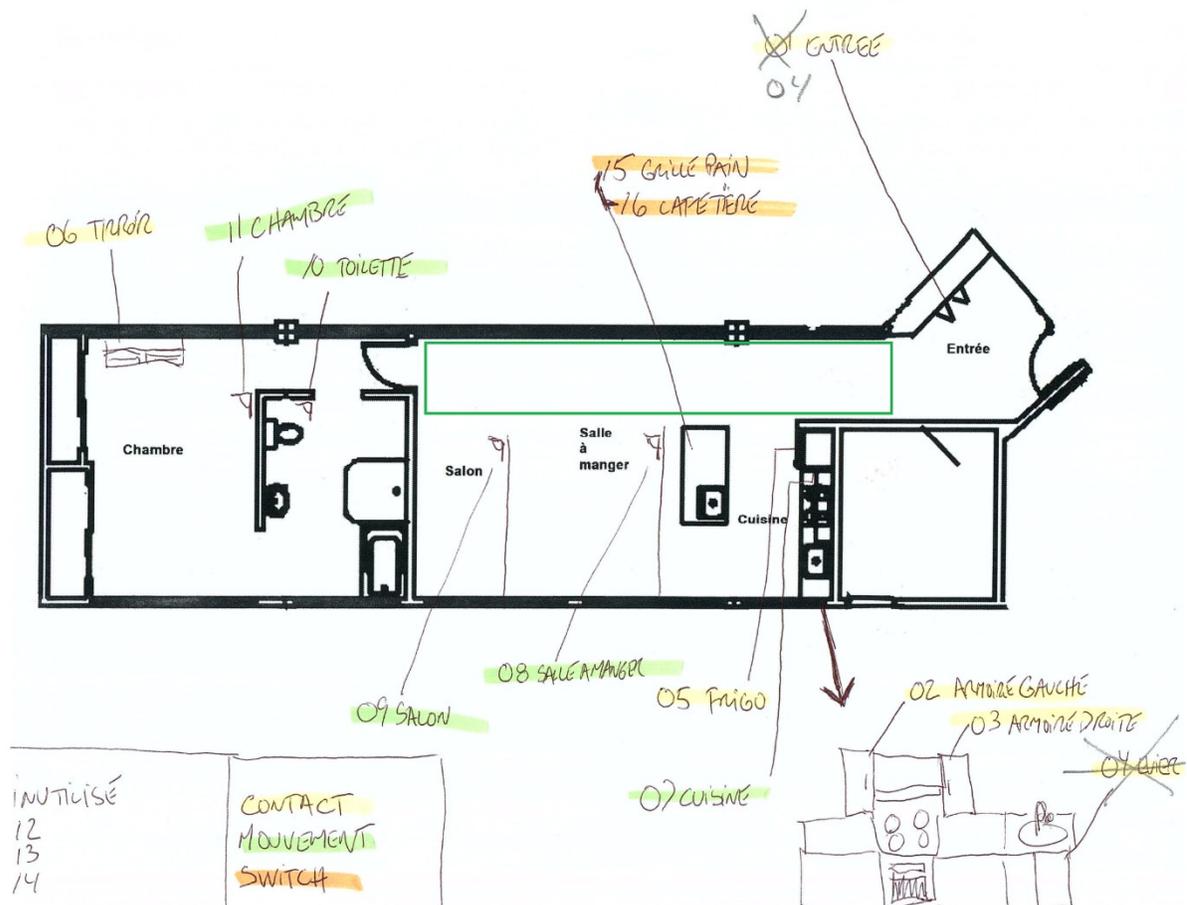


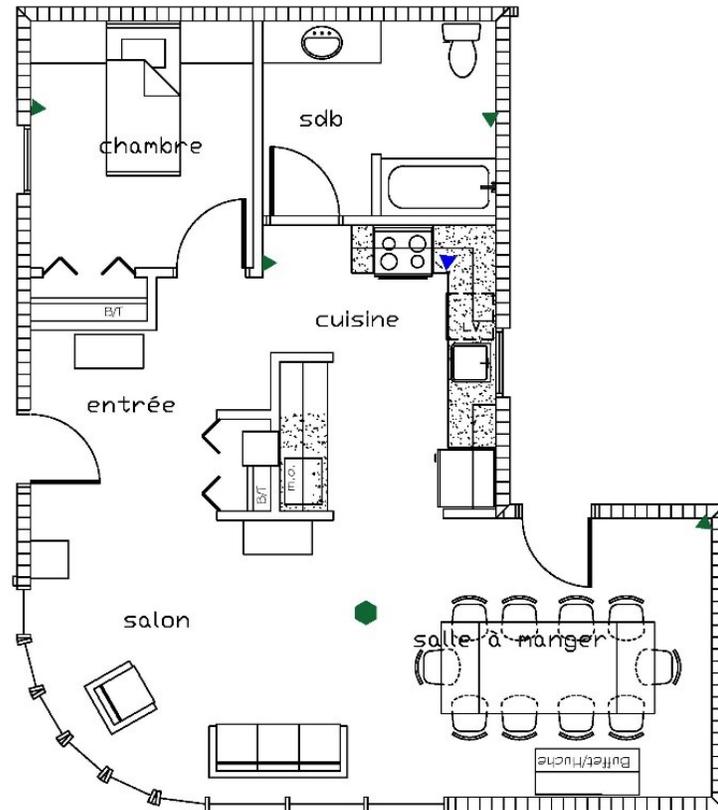
Figure 8 – Plan du laboratoire du CRIUGM

3.1.3 Multimédia

Dans le but de filmer les expérimentations, des caméras fixes sont installées pour pouvoir observer le participant, peu importe où il se trouve (Figure 9). Nous avons donc une caméra dans la chambre, une dans la salle de bain et une dans la salle à manger qui couvre aussi le salon. Comme la plupart des tâches se passent dans la cuisine, nous avons pris soin d'y installer trois caméras qui ont chacune un angle de vue complémentaire. Nous avons une caméra au plafond située entre le salon et la salle à manger qui est orientée vers la cuisine, une caméra qui regarde la zone de la cuisinière, et un caméscope qui regarde la zone où sont placés la cafetière et le grille-pain. L'entrée de l'appartement n'est pas couverte, car nous avons affaire à des contraintes de budget, ainsi que des stocks limités chez notre magasin d'électronique local. Ce n'est pas très dérangent, car le participant ne reste généralement dans l'entrée que quelques secondes.

La technologie IP présente sur la plupart des caméras fixes de l'appartement a été grandement utile pour la retransmission vidéo. En effet, grâce au logiciel VLC [33] utilisé pour enregistrer le flux vidéo, il est possible de rediriger le flux sur le réseau Wi-Fi et de le regarder sur un ordinateur portable en direct. Ainsi, lors des expérimentations, un ordinateur portable était placé derrière le divan pour que l'expérimentateur puisse observer en direct les activités du participant lorsque celui-ci se trouve dans la salle de bain ou dans la chambre.

En plus des caméras, des micros sont intégrés au plafond de la salle à manger, cuisine, chambre et salle de bain. Les caméras IP et le caméscope ont des micros intégrés, mais pour être certain de ne pas avoir de problème d'acquisition audio ou de qualité du son, il n'était pas très compliqué d'utiliser en plus les micros déjà installés.



Légende	
Caméra Sony 360°	■
Caméra IP D-Link	▶
Caméscope sur le comptoir	▲
Echelle (en mm)	1000

Figure 9 – Emplacement des caméras fixes

Finalement, dans l'appartement du CRIUGM, aucun micro ni caméra n'est fixé dans l'appartement. L'acquisition multimédia se fait plutôt à l'aide d'un caméscope à l'épaule de l'expérimentateur. Celui-ci se place le long du corridor, en dehors de la portée des capteurs de mouvement. L'expérimentateur suit les déplacements du participant en restant à l'écart, sans le déranger.

3.2 Architecture

L'architecture du système d'acquisition de données (Figure 10) se décompose en quatre sous-réseaux : celui des caméras IP, de la tablette Android, des capteurs filaires et des capteurs sans fil.

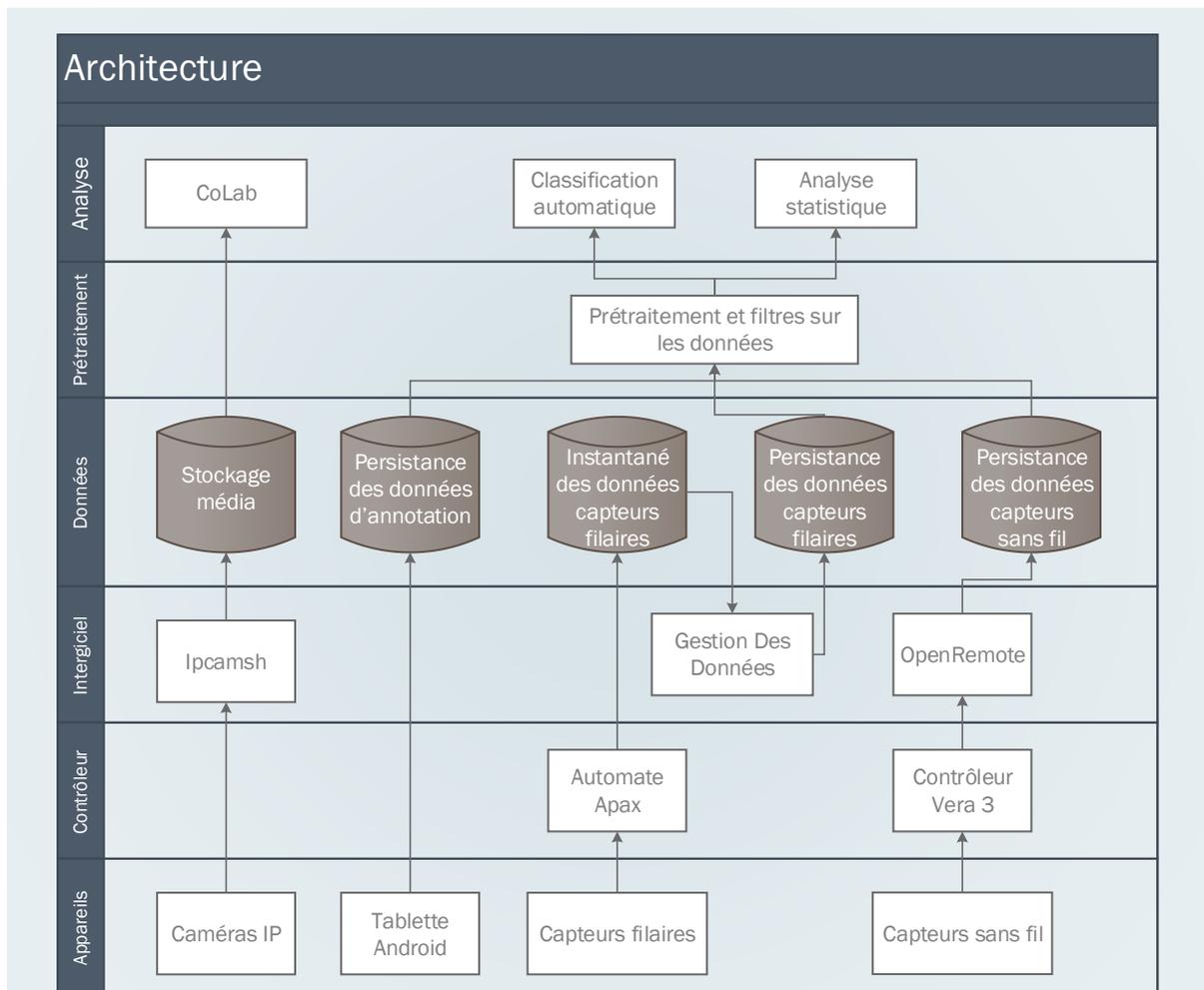


Figure 10 – Architecture du système d'acquisition de données

Commençons avec les caméras IP. Ce sont elles qui acquièrent les vidéos d'expérimentation. Ensuite, au niveau de l'intergiciel se trouve le script Ipcamsh qui sert à démarrer l'acquisition des vidéos et qui permet de superviser l'état de l'acquisition grâce aux informations qu'il transmet : l'utilisation de l'UCT, de la mémoire vive, du disque dur et le débit réseau et ce,

pour chaque instance du logiciel VLC, donc pour chaque caméra. Les vidéos sont ensuite stockées sur un serveur du laboratoire DOMUS, pour enfin être visionnées dans l'application web CoLab.

Continuons avec la tablette Android. Les données d'annotation sont stockées dans une base de données (BD) SQLite [34] intégrée à la tablette. Une fois les expérimentations terminées, les données seront extraites de la tablette et stockées sur le serveur du DOMUS en format .bd et .csv, prêtes à être prétraitées, puis analysées.

Pour les capteurs filaires, les valeurs des capteurs sont mises à jour dans une table de la BD Microsoft SQLServer [35] du DOMUS à chaque 200ms. Cela représente un instantané de l'état de tous les capteurs à l'instant présent. L'automate Apax (modèle APAX-6572-AE [36]) s'occupe d'acquérir ces valeurs et de les envoyer à la BD. Ensuite, l'application **Gestion Des Données** va quérir ces nouvelles valeurs et les stocke de façon permanente dans une autre table de la même BD. L'application répète cette opération à chaque seconde, le système ne pouvant garantir d'être en mesure de compléter l'opération plus rapidement de façon constante. Nous avons donc une nouvelle insertion dans la table de persistance des données chaque seconde d'expérimentation. Enfin, ces données seront prétraitées, puis analysées.

Pour les capteurs sans-fil, les capteurs s'occupent eux-mêmes d'avertir le contrôleur Vera 3 [37] d'un changement d'état. Ils communiquent ensemble à l'aide de la technologie sans fil Z-Wave [11]. Le contrôleur ne garde en mémoire que la valeur courante de l'état des capteurs. Pour contrer ce problème, l'intergiciel OpenRemote [38] recueille les nouvelles valeurs du contrôleur, qui seront stockées dans une BD PostgreSQL. OpenRemote utilise la technique de *HTTP long polling*, ce qui consiste à faire une requête HTTP au contrôleur demandant l'état de chaque capteur. Le contrôleur ne répondra que lorsqu'il aura une information à transmettre, donc une mise à jour de l'état d'au moins un capteur. Tant qu'il n'y a pas de mise à jour, la requête est mise en attente. Puis, lorsqu'OpenRemote reçoit une réponse, il relance une nouvelle requête du même type pour connaître les prochaines mises à jour sans perdre de temps. La BD des capteurs sans fil est différente de celle du réseau de capteurs filaires pour garder les deux réseaux totalement indépendants et utiliser le plus de technologies libres et gratuites

possible pour un futur déploiement du système sur de nombreux sites d'expérimentations. Finalement, à l'instar des données de la tablette et du réseau de capteurs filaires, celles-ci seront prétraitées et analysées ultérieurement.

Chapitre 4

Annotation des données

Pour faire de la classification automatique supervisée, il est nécessaire d'ajouter la classe associée à chaque sous-ensemble des données. Le concept d'apprentissage supervisé implique aussi que nous pouvons bonifier les données d'apprentissage d'informations pertinentes sur le contexte de ces données (des annotations), ce qui peut aider les algorithmes d'apprentissage à mieux classer les données. C'est pour cela que nous avons, dans le cadre de ce projet pilote, conçu l'Application Projet Dépistage (APD). C'est une application pour tablette Android utilisée pendant une expérimentation et recueillant des informations sur la durée des tâches accomplies par le participant. De plus, l'application web CoLab, qui a aussi été conçue dans le cadre de ce projet principalement pour le visionnement des vidéos d'expérimentation, offre l'opportunité aux spécialistes collaborant sur ce projet d'ajouter des annotations et des observations de grille d'analyse comportementale. Ces annotations viendront aussi bonifier les données de capteurs. Ainsi, nous détaillerons ici le but et les fonctionnalités de ces applications.

4.1 Application Projet Dépistage (APD)

L'Application Projet Dépistage (APD) est une application grâce à laquelle nous pouvons connaître la durée de chaque tâche et le temps total d'une expérimentation. Nous expliquerons dans quel but nous avons conçu cette application, ses fonctionnalités et les améliorations possibles.

4.1.1 But de l'application

Dans le cadre des expérimentations réalisées, nous avons besoin d'un moyen de savoir précisément l'heure exacte de début et de fin à laquelle chaque tâche a été amorcée, mise en pause ou terminée, pour savoir aisément combien de temps a pris chaque participant pour

exécuter chaque tâche. Ces annotations permettront d'analyser les données en les séparant par tâche (par exemple, ne regarder les données de capteurs que lors de l'exécution de la tâche de la préparation de repas), ce qui pourrait limiter le bruit causé par le grand nombre de capteurs ou par les autres tâches. Enfin, ces annotations permettront ainsi de vérifier si les participants atteints de troubles cognitifs prennent significativement plus de temps pour accomplir les tâches que les autres, et donc si cette variable est un facteur discriminant. Une autre information importante qu'il nous faut connaître est l'heure exacte du début et de la fin de l'expérimentation et donc, la durée totale de l'expérimentation. Cette dernière sera utile pour synchroniser l'heure de début et de fin entre nos diverses sources d'acquisition de données.

Bien entendu, l'expérimentateur aurait pu noter les temps à la main, à l'aide d'une montre, mais observer les faits et gestes d'un participant demande une trop grande attention pour accomplir tout cela manuellement. C'est pourquoi nous avons besoin d'une solution logicielle, où l'accent serait mis sur l'utilisabilité et l'efficacité, pour assurer la fidélité de la prise de données. Cette solution devait être utilisable pendant l'expérimentation, ainsi le support tablette a été choisi. Nous avons donc développé une application Android [13], qui est une plateforme bien connue pour téléphones et tablettes. Développer sur Android est gratuit, comparativement à d'autres plateformes concurrentes. Finalement, le laboratoire DOMUS avait déjà à sa disposition une tablette Nexus 7 que nous pouvions utiliser pour notre projet.

4.1.2 Fonctionnalités

Comme l'expérimentateur est occupé à observer les moindres faits et gestes du participant, l'interface de l'application se doit d'être la plus simple à utiliser possible en réduisant les risques d'erreur de manipulation. Ainsi, les boutons système d'Android ont été cachés – il faut glisser le doigt à partir du bas pour les faire réapparaître. La navigation entre les pages de l'application (navigation *back* et *up*) est validée par une demande de confirmation. Il est donc difficile de quitter l'écran principal de l'application par erreur. Une confirmation est aussi demandée pour indiquer qu'une tâche est terminée, et l'utilisateur ne peut plus redémarrer cette tâche par la suite.

Voyons maintenant toutes les fonctionnalités de l'application ainsi que quelques captures d'écrans. Lors du lancement de l'application, une page de démarrage s'affiche (Figure 11). Elle permet de nommer un scénario d'expérimentation et pourra le lancer au moment voulu. Lorsque l'utilisateur appuie sur **Démarrer un nouveau scénario**, le temps à la seconde près est enregistré comme étant le début de l'expérimentation, puis l'écran principal s'affiche.

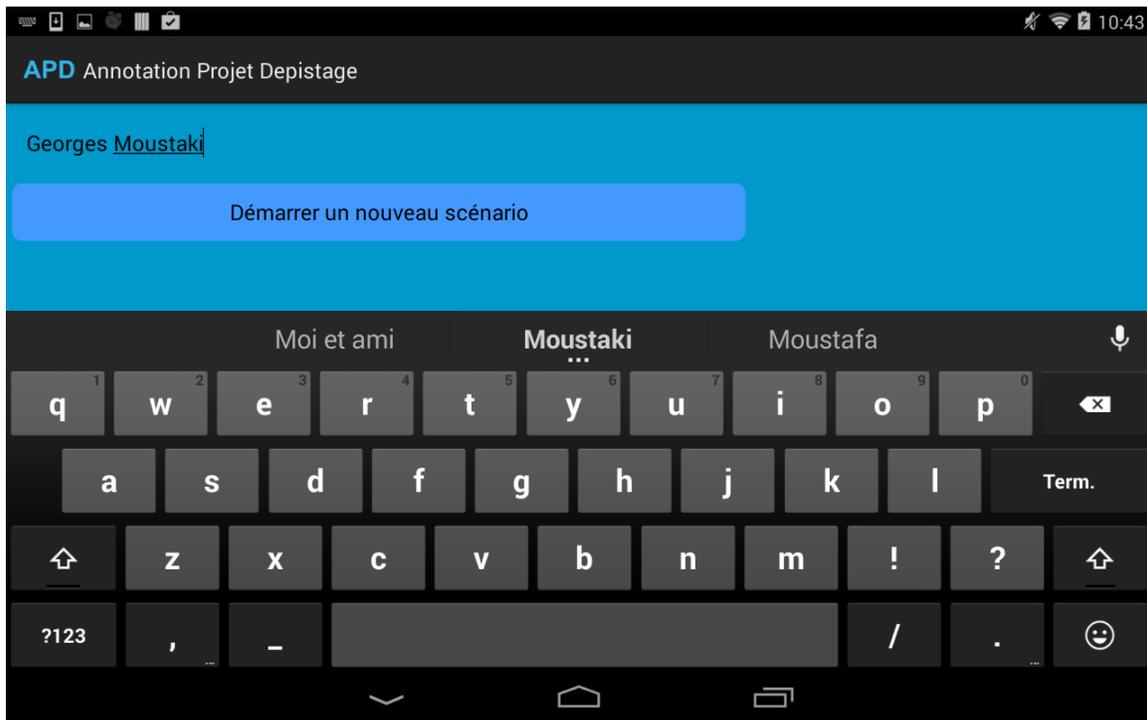


Figure 11 – Écran de démarrage avec affichage du clavier Android

L'application affiche une liste de tâches à gauche et les sous-tâches à droite après avoir choisi une tâche parente (Figure 12). Par exemple, « préparer un repas léger » se décompose en « préparer un café », « préparer deux rôties », « dresser la table », etc. Une arborescence à deux niveaux (tâche parente avec enfants) a donc été créée pour chaque tâche. L'expérimentateur peut changer l'état des sous-tâches. S'il appuie sur une tâche à droite, la tâche démarre ou est mise en pause, selon son état actuel. Appuyer sur le bouton carré termine la tâche. Voici la liste des couleurs associées à l'état des sous-tâches :

- **Grise** : la tâche n'est pas débutée.
- **Verte** : la tâche est démarrée. Le temps de début est enregistré.
- **Jaune** : la tâche est mise en pause. Appuyer une seconde fois la redémarre.
- **Rouge** : la tâche est terminée. On ne peut plus la redémarrer.



Figure 12 – Écran principal avec sous-tâches dans divers états

En cours de route, l'expérimentateur peut rajouter une nouvelle sous-tâche directement dans l'application s'il se rend compte que le participant effectue une activité non demandée ou décide de refaire une tâche déjà complétée (Figure 13). L'ajout se fait en simplement en appuyant sur le bouton « Ajouter une autre tâche »⁹. La nouvelle sous-tâche est ajoutée dans la tâche courante, à la suite des sous-tâches déjà présentes. Elle est identifiée par un numéro de sous-tâche et l'heure à laquelle elle a été ajoutée. L'expérimentateur doit ensuite noter les

⁹ Qui devrait plutôt être renommé : « Ajouter une autre sous-tâche ».

détails sur papier si nécessaire. Cette solution offre une rapidité de création nécessaire pendant une expérimentation.



Figure 13 – Écran principal avec une nouvelle sous-tâche ajoutée

Dans la barre en haut de l'écran, on voit le nom de l'application et le nom du scénario du scénario à gauche, et un bouton sous forme de trois points alignés verticalement à droite. Appuyer sur ce bouton ouvre un menu secondaire. La seule option disponible de ce menu est de terminer le scénario. L'emplacement inhabituel de ce bouton a été pensé pour limiter les risques que l'expérimentateur termine le scénario par erreur.

Lorsque l'expérimentation est terminée, on passe à l'écran d'évaluation des tâches. Cet écran sert à noter le participant sur ses aptitudes à accomplir les tâches demandées. On y voit à gauche toute la liste des tâches. Au centre, le temps utilisé pour accomplir la tâche sélectionnée (en vert) est affiché et à droite la note attribuée à cette tâche (Figure 14). Lorsque l'on appuie sur une tâche, une fenêtre s'ouvre demandant d'attribuer une note à la tâche sélectionnée (Figure 15). Cette échelle de notation est tirée du Profil des AVQ [6].

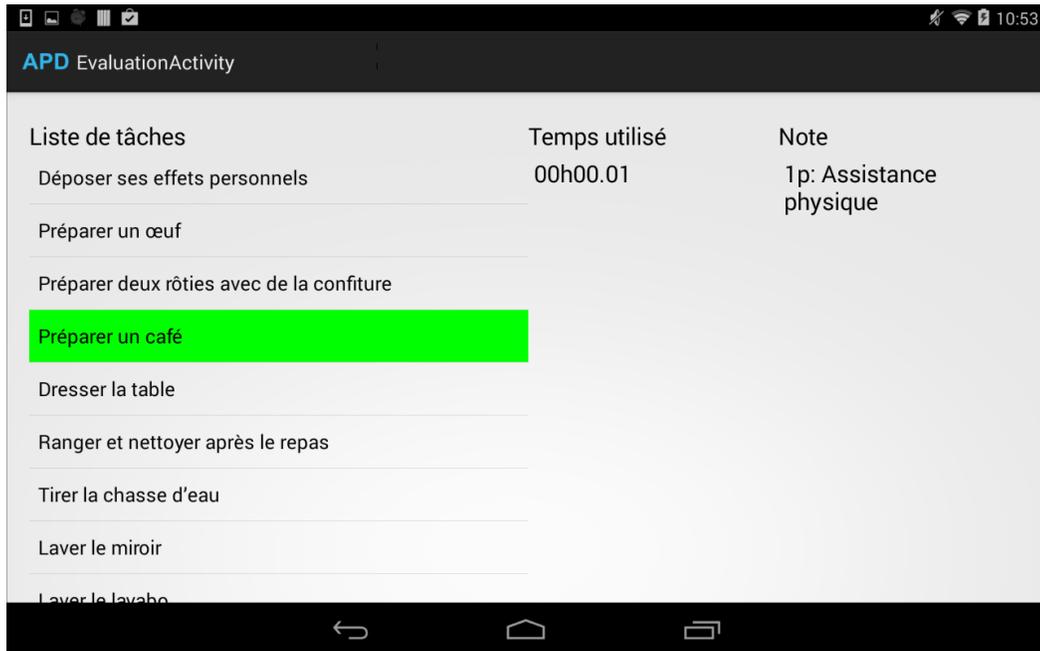


Figure 14 – Écran d'évaluation des tâches avec la tâche « Préparer un café » sélectionnée

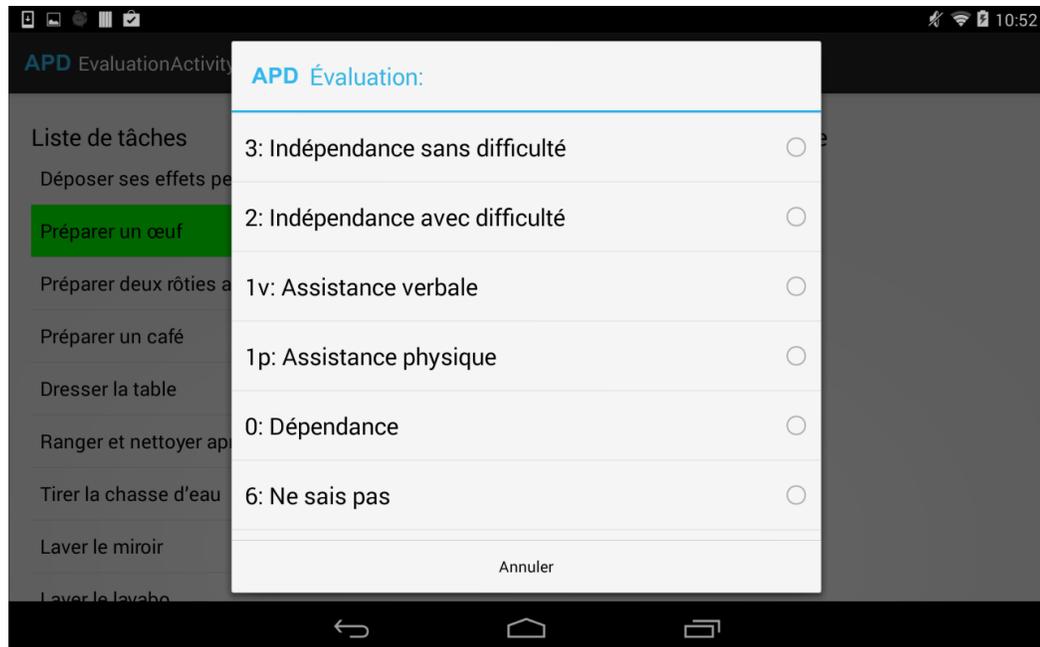


Figure 15 – Fenêtre demandant d'attribuer une note à la tâche « Préparer un café » que l'on vient de sélectionner

4.1.3 Stockage des données

L'application dispose d'une base de données SQLite [34] locale (dans la tablette) pour stocker certaines données en cours d'utilisation et en sauvegarder d'autres pour consultation ultérieure. Nous avons choisi SQLite, car c'est le système de gestion de base de données par défaut sur Android. Les données les plus importantes (l'heure exacte à laquelle chaque tâche est démarrée, mise en pause ou terminée) sont aussi sauvegardées directement en format CSV pour une lecture aisée dans Excel. Rien n'empêche d'exporter en CSV à partir de la BD SQLite, mais le faire directement sauve quelques étapes. Pour extraire les données de la tablette, la méthode utilisée est d'accéder à sa mémoire interne en la branchant par câble USB dans un ordinateur avec Android Studio installé. Consultez l'Annexe C.1 pour connaître les commandes à exécuter une fois la tablette branchée.

Voici la liste des tables de la BD :

- *Task* : table conservant la liste des tâches à faire, la tâche parente pour chaque sous-tâche et dans quel état chaque tâche se trouve à l'instant présent parmi les suivants : NOT_STARTED, ACTIVE, PAUSED, FINISHED.
- *Scenario* : table servant à prendre en note l'heure de début et de fin du scénario et le nom du scénario.
- *Evaluation* : table recueillant les évaluations données par l'expérimentateur selon les performances du participant. Les tâches sont notées après avoir terminé le scénario.
- *Recording_task* : à chaque fois que l'on change l'état d'une sous-tâche dans l'application, une nouvelle ligne est créée dans cette table. Celle-ci enregistre donc tous les changements d'état des tâches et le moment précis où cela a été fait.

4.1.4 Perspectives

L'application avait été réfléchi initialement pour les besoins du projet de dépistage précoce de l'Alzheimer, mais elle pourrait être utilisée pour d'autres expérimentations ultérieurement, avec quelques modifications mineures. Pour l'instant, la liste des tâches et l'échelle de notation sont ajoutées directement dans le code. Quoiqu'il soit facile de modifier ces informations, il serait préférable, afin de simplifier la procédure, que celles-ci puissent être fournies à l'application de façon externe, et donc, qu'elles soient interchangeable, par exemple dans un document de type XML.

4.2 CoLab

CoLab est un logiciel de collaboration multidisciplinaire, de visionnement de vidéos d'expérimentation et d'annotation de données développé dans le cadre de notre projet par mon collègue Jules Randolph. Il s'agit d'une application web conçue à l'aide des technologies Meteor [39] et JavaScript. Elle a été créée pour ce projet-ci, mais a été conçue de façon à être assez générique pour d'autres projets du DOMUS ou d'autres laboratoires qui utilisent des caméras IP et des capteurs pour surveiller le comportement humain. CoLab a aussi été pensé pour être simple, utile et convivial pour tous les utilisateurs, qu'ils soient informaticiens ou ergothérapeutes. Il est important de noter que, dans le développement de cette application, ma contribution se restreint à l'élicitation des exigences et aux tests de l'application.

4.2.1 But de l'application

Les ergothérapeutes et autres chercheurs dans le domaine de la santé qui collaborent sur le projet avaient comme besoin de pouvoir consulter les vidéos d'expérimentation, de noter leurs observations et d'appliquer leur grille d'analyse. Les vidéos devant rester confidentielles, nous ne pouvions pas partager les fichiers sources à l'aide d'un serveur FTP ou d'un service

inonuagique; le partage non autorisé aurait été trop facile à faire. C'est pourquoi l'idée d'une application web sécurisée avec comptes utilisateurs a été retenue.

Le visionnement sur demande des vidéos d'expérimentation est aussi très utile pour la validation du système d'acquisition de données hétérogènes. En effet, une majorité de capteurs sont en redondance avec les caméras. Il est donc possible, si l'on doute d'un faux déclenchement ou d'une quelconque dysfonction d'un capteur, de regarder la vidéo qui est en redondance avec le capteur, au même moment où celui-ci est censé se déclencher, et vérifier si le participant a bel et bien activé le capteur. Par exemple, si on soupçonne le capteur de mouvement du salon de s'être activé à un moment donné en raison d'un déplacement de l'expérimentateur et non du participant (comme prévu), nous n'avons qu'à regarder la vidéo au même moment pour vérifier ce qui a fait déclencher le capteur : le participant ou l'expérimentateur. Cette redondance a été grandement utile pour valider nos données de capteurs sans fil lors du prétraitement des données, détaillé au Chapitre 5.

4.2.2 Fonctionnalités

Nous exposerons les fonctionnalités de l'application ainsi que quelques captures d'écrans. Nous ne décrirons que les fonctionnalités pertinentes pour le présent mémoire. Le reste sera couvert dans le mémoire de mon collègue Jules Randolph.

Une fois que l'utilisateur est authentifié et que le projet est choisi, l'utilisateur a accès à la liste des expérimentations, triées par nom d'expérimentation, par date ou par durée (Figure 16). À gauche, dans la colonne « collaboration », on aperçoit quatre symboles associés à une expérimentation donnée. Le premier indique si l'expérimentation est publiée ou non, donc accessible aux collaborateurs qui ne sont pas administrateurs. Le deuxième indique si l'expérimentation a été validée par l'équipe technique (la présence des vidéos a été confirmée), tandis que le troisième indique si elle a été validée par l'équipe clinique (respecte les critères cliniques). Une expérimentation serait invalidée du côté technique par exemple si l'acquisition vidéo avait flanché et qu'aucune vidéo n'était disponible. Du côté clinique, une

expérimentation serait invalidée si le participant avait été malade et n'avait pas pu compléter l'expérimentation. Finalement, le quatrième symbole représente si l'ensemble des vidéos ont été synchronisés manuellement entre eux, tâche nécessaire, car les caméras ne pouvaient pas être démarrées toutes au même moment. Enfin, on choisit l'expérimentation dont on veut consulter les vidéos en cliquant sur son nom.

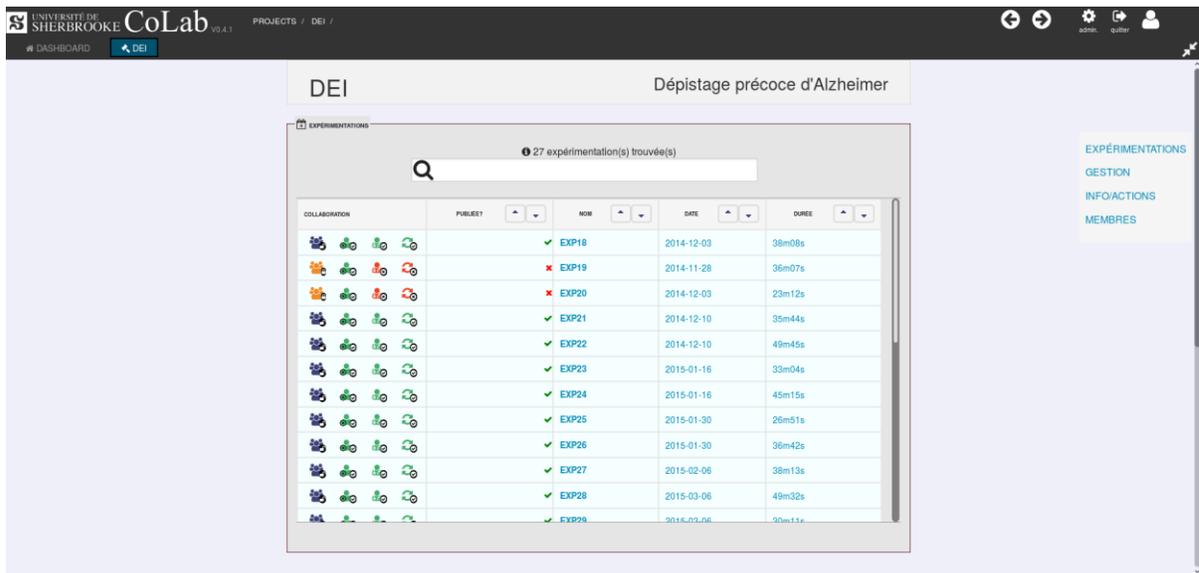


Figure 16 – Choix d'une expérimentation du projet DEI

Comme le but premier de l'écran suivant, présenté à la Figure 17, est de visualiser une vidéo, l'interface de CoLab est constituée de la vidéo au centre et au-dessous de la vidéo se trouve une barre contenant les contrôles vidéo habituels comme le bouton « Démarrer / Pause » (symbole de triangle orienté vers la droite), le contrôle du volume, la vitesse de lecture (variant de 0.5x à 4x) et la barre de progression. Le menu de gauche permet de choisir la source d'acquisition affichée (il y avait six caméras pour le projet DEI). Au bas de l'écran, nous avons l'explorateur temporel où s'affichent des bandes de couleur représentant la durée des tâches à accomplir. Par exemple, on voit que la tâche « Laver le miroir » (en violet) a été commencée vers 7m40 et s'est terminée vers 8m45. Si l'on clique sur cette bande de couleur (Figure 18), le nom de la tâche et les temps exacts s'affichent, ainsi que l'option de modifier ces informations et de démarrer la vidéo au début de cette tâche.



Figure 17 – Écran principal de CoLab

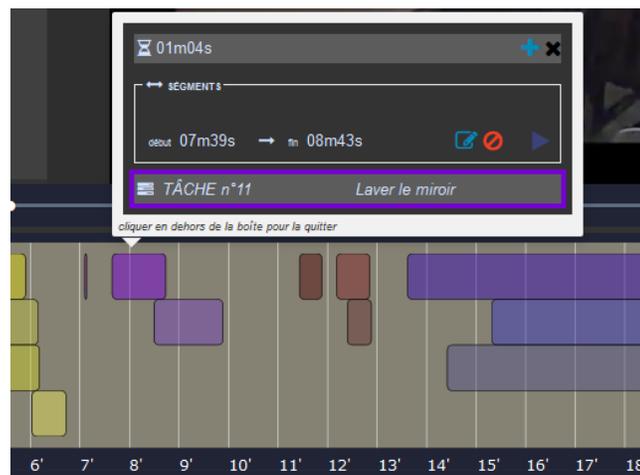


Figure 18 – Détails d'une tâche dans l'explorateur temporel

La section « paramètres » du projet (Figure 19), que l'on atteint à partir de la page principale du projet (Figure 16), permet d'ajouter et de modifier la liste de tâches. Cette fonctionnalité permet à CoLab d'adapter la liste de tâches à d'autres projets qui utiliseraient une arborescence en deux niveaux comme le projet DEI. La personnalisation du nombre de niveaux n'a pas été implémentée dans cette version de CoLab, mais pourrait l'être dans le futur. Sur chacune des

étiquettes représentant une tâche, on retrouve deux symboles : un pour modifier, l'autre pour supprimer la tâche.

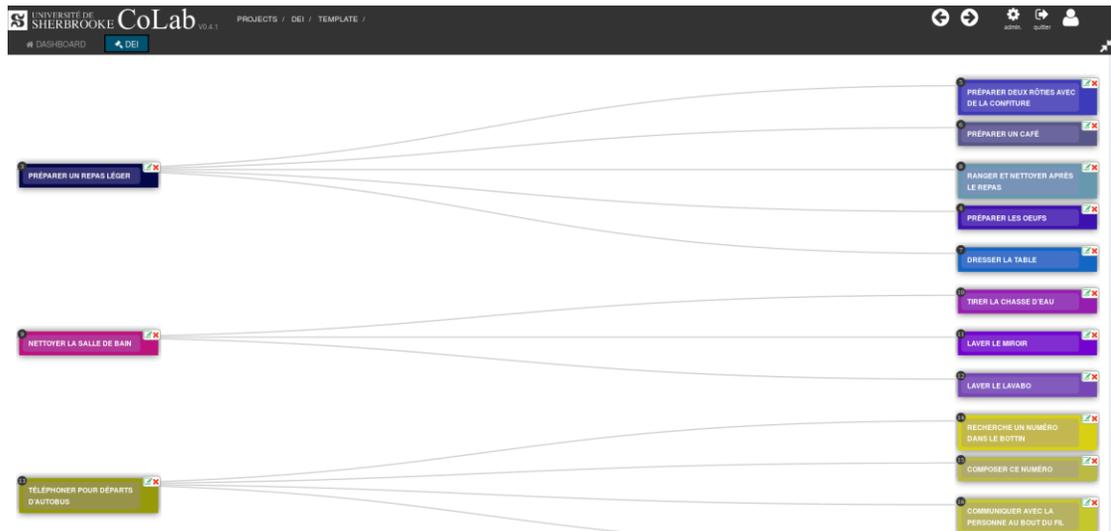


Figure 19 – Liste de tâches et choix des couleurs d’affichage

Si l’on décide de modifier la tâche (Figure 20), nous pouvons personnaliser le nom de la tâche, la couleur qui la représente et la tâche parente, le cas échéant.

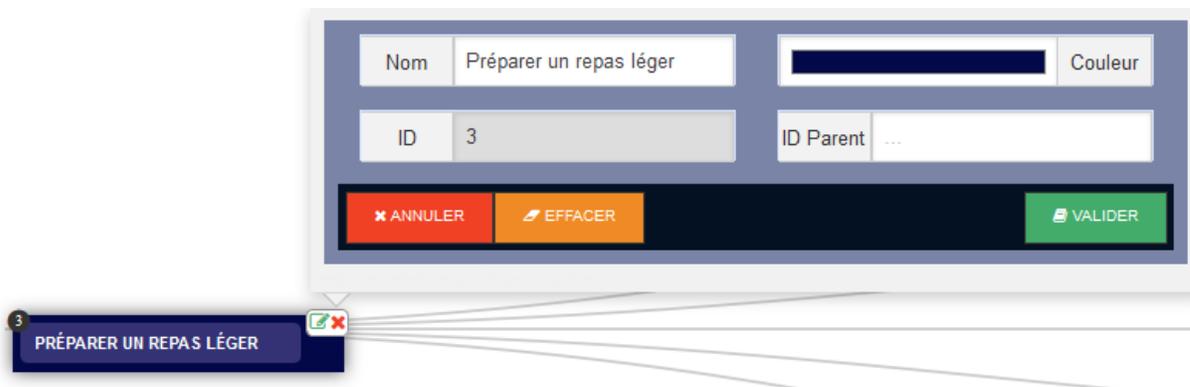


Figure 20 – Modification d’une tâche

Finalement, il est aussi possible d’intégrer et de modifier la grille d’analyse comportementale utilisée par les spécialistes (Figure 21). Cette grille est, pour ce projet, divisée en deux catégories : erreur et assistance, qui se subdivisent en cinq sous-catégories. À chaque fois qu’un

utilisateur de CoLab ajoute une observation (à l'aide de l'interface illustrée précédemment à la Figure 18), il peut spécifier s'il s'agit d'une erreur ou d'une assistance, et son type. Un nombre plus élevé d'erreurs ou d'assistance est généralement signe d'un trouble cognitif plus marqué [6]. C'est donc une variable discriminante. Ainsi, le nombre d'annotations de chacune des dix sous-catégories pourrait être cumulé par expérimentation et être ajouté aux données d'apprentissage pour aider à discriminer les participants parmi les trois classes : sujets sains, TCL et DTA.

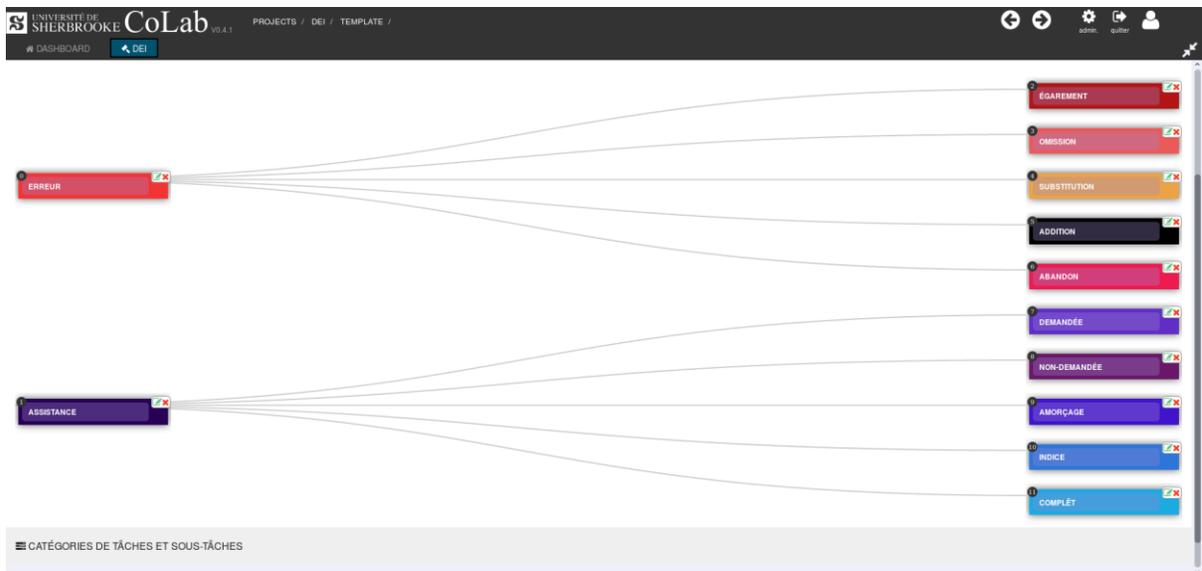


Figure 21 – Grille d'analyse comportementale utilisée par les spécialistes

4.2.3 Perspectives

L'application CoLab est, jusqu'à maintenant, à l'étape de prototype utilisé dans le cadre du projet DEI. Ainsi, des améliorations et de nouvelles fonctionnalités seraient à prévoir pour de futurs projets. À la Figure 22, nous pouvons observer une maquette proposant un nouveau design pour CoLab, ainsi que deux nouvelles fonctionnalités majeures : la saisie de verbatim (à droite) et l'affichage des annotations d'observation des spécialistes, de saisie de verbatim et d'événement de capteurs. Dans une prochaine version, nous aurons donc accès en un coup d'œil à toutes les annotations ajoutées à la vidéo par les collaborateurs ayant déjà analysé la

vidéo, ainsi que les déclenchements de capteurs pendant l'expérimentation. La liste de capteurs à afficher serait personnalisable, car un grand nombre de capteurs comme au laboratoire DOMUS encombrerait trop l'interface.

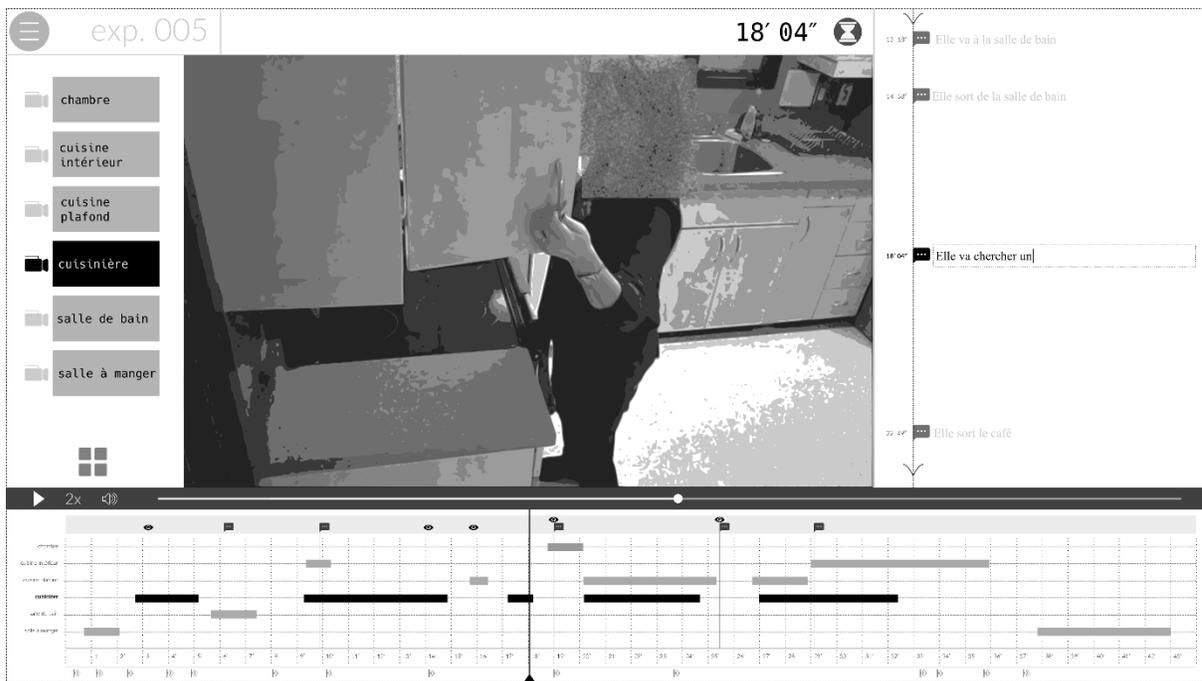


Figure 22 – Maquette du design proposé de CoLab

4.3 Conclusion

Nous avons vu dans ce chapitre deux applications développées dans le cadre de ce projet pour nos besoins. L'application tablette APD a pour but d'accumuler des données supplémentaires, les temps de début et de fin des tâches, qui viendront bonifier les données de capteurs lors de l'apprentissage supervisé. L'application web CoLab a été conçue pour répondre à différents besoins. Nous avons décrit ceux permettant de colliger les observations des spécialistes, de bonifier les données de capteurs, ainsi que d'offrir la possibilité de vérifier l'intégrité de nos données de capteurs grâce aux vidéos d'expérimentation. Tout cela dans le but d'avoir des données d'apprentissage fiables contenant le plus d'informations connexes et pertinentes possible.

Chapitre 5

Expérimentations et prétraitement des données

Pour mettre à l'épreuve notre infrastructure d'acquisition de données et tester nos hypothèses quant à la faisabilité d'utiliser ces données pour le dépistage précoce de la DTA, des expérimentations se sont déroulées en laboratoire avec des personnes âgées, en respectant le protocole établi au Chapitre 2. Une grande quantité de données de différents types a été recueillie. Enfin, les données ont été prétraitées afin de pouvoir les utiliser.

5.1 Informations démographiques des participants

Le recrutement à Sherbrooke a été réalisé à travers des contacts personnels et par la Clinique de mémoire du CSSS-IUGS. Une liste de critères (Annexe D) devait être respectée pour qu'une personne soit recrutée dans le projet. Chaque participant devait avoir 65 ans ou plus et être suffisamment en santé et mobile pour accomplir les tâches. Le recrutement de participants ayant un TCL ou une DTA qui respectent les critères de sélection a été considérablement ardu, car peu de gens étaient volontaires et aptes. C'est principalement pour cela qu'il n'y a eu que 4 participants DTA à Sherbrooke.

Lors des expérimentations, nous avons reçu, au Laboratoire DOMUS, 24 participants répartis en 3 groupes:

- 10 sujets sains ne présentant aucun signe de déclin cognitif diagnostiqué;
- 10 sujets ayant un TCL (ou *Mild Cognitive Impairment* en anglais) diagnostiqué;
- 4 sujets atteints d'une DTA au stade léger.

Au laboratoire du CRIUGM, ce fut 36 personnes qui ont participé aux expérimentations (16 sujets sains, 16 sujets atteints d'un TCL, 4 sujets atteints d'une DTA).

Rappelons que tous les participants ont passé des tests cognitifs et questionnaires qui ont servi essentiellement à vérifier les capacités cognitives de ceux-ci et ainsi s'assurer que les participants respectent bien les critères d'inclusion.

5.2 Déroulement et problèmes lors des expérimentations

Cette section détaillera le déroulement et les problèmes survenus lors des expérimentations au laboratoire DOMUS. Cela permet de détailler les avantages et les obstacles d'une expérimentation en regard des habitats intelligents. Les expérimentations se sont échelonnées sur onze mois, de septembre 2014 à août 2015. Des prétests ont été réalisés pour tester le protocole et le système d'acquisition de données. Les participants observés étaient des étudiants. Pour ces prétests, et pour les deux premières expérimentations en septembre 2014, nous avions des caméscopes sur pied au lieu de caméras IP (voir Chapitre 3). Les caméscopes ont été remplacés ensuite par des caméras IP, car :

- Les caméscopes s'éteignaient avant la fin de l'expérimentation, car la capacité des piles fournies était insuffisante;
- Les caméscopes étaient empruntés à l'Université et ne pouvaient pas être loués pour toute la durée des expérimentations;
- Les caméras fixées au mur offrent une solution permanente et sont contrôlables à distance avec un flux vidéo retransmissible.

L'installation et la configuration des caméras, ainsi que le développement du script **Ipcamsh**¹⁰ pour démarrer l'acquisition de vidéo à distance a demandé plus d'un mois. Une mise à niveau sur les technologies permettant le visionnement et l'enregistrement de flux vidéo par réseau a été nécessaire. Les expérimentations ont repris leur cours normal à la fin novembre 2014.

¹⁰ La version initiale du script a été codéveloppée avec Jules Randolph. Il est responsable de la version finale.

Par la suite, à la mi-décembre, l'acquisition de données de capteurs sans fil a rencontré un problème majeur. Dans l'infrastructure du réseau de capteurs sans fil, le contrôleur Vera 3 est en communication avec l'intergiciel OpenRemote pour l'informer d'un changement d'état d'un capteur. Lors de deux expérimentations, la communication entre ces deux composants a été coupée pendant plus de dix minutes. Comme le contrôleur Vera ne garde pas en mémoire l'état des capteurs, il y a donc eu une perte de données conséquente. La cause a été identifiée début janvier 2015. En effet, le contrôleur Vera possède une interface web pour vérifier l'état des capteurs et les configurer, à laquelle on se connecte à l'aide d'un navigateur lors des vérifications précédant chaque expérimentation. Or, il a été remarqué que le contrôleur Vera est incapable de maintenir correctement une connexion avec OpenRemote en même temps qu'une autre connexion avec un navigateur. La connexion va indéniablement lâcher d'un côté ou de l'autre. Il est donc nécessaire de fermer toute connexion superflue avec le contrôleur Vera avant de démarrer l'acquisition de données. Enfin, après cela, aucun problème technique majeur n'est survenu pour le reste (la plupart) des expérimentations.

Une fois les expérimentations terminées, lors de l'analyse des données, il a été remarqué que le nombre de déclenchements de certains capteurs de mouvement sans fil semblait anormal. En effet, les capteurs du salon et de la salle à manger se déclenchent plus souvent que le nombre de fois où le participant est vraiment entré dans la salle correspondante. Cela a été vérifié en comparant la présence par vidéo d'expérimentation et les horodatages des capteurs. Ce problème provient du chevauchement des zones de captation des capteurs. Par exemple, lorsque le participant entre dans la salle à manger, le capteur de mouvement de la salle à manger et celui du salon se déclenchent en même temps. De plus, comme ces capteurs ne peuvent pas se déclencher à une fréquence plus rapide qu'à chaque dix secondes (une limitation du capteur), il est possible, par exemple, que le capteur du salon se déclenche par erreur lorsque le participant entre dans la salle à manger, mais ne détecte pas son passage cinq secondes plus tard. Pour corriger le problème, nous avons croisé tous les déclenchements de capteurs de mouvement avec ce qui se passe dans les vidéos et supprimé les déclenchements fautifs. En conclusion, ces problèmes et manipulations nuisent considérablement à la qualité des données

issues des capteurs sans fil, comparativement aux capteurs filaires avec lesquels nous n'avons eu aucun problème.

5.3 Données récoltées

Une multitude de données provenant de plusieurs sources ont été récoltées pour ce projet: capteurs, caméras, annotations manuelles de la durée des tâches effectuées et observations des spécialistes. Cette section décrira les données et comment elles ont été prétraitées.

5.3.1 Description des données

Dans cette section, chaque type de données recueillies lors des expérimentations sera détaillé.

5.3.1.1 Données du réseau de capteurs filaires

Le système d'acquisition de données du réseau de capteurs filaires du laboratoire DOMUS a été conçu pour enregistrer l'état de tous les capteurs chaque seconde, qu'il y ait eu changement d'état ou non. Cela représente ainsi un nouveau tuple (vecteur) dans la table d'enregistrement de la BD à chaque seconde d'expérimentation. Cette table (Tableau 4) est composée : d'un ID unique pour chaque enregistrement de donnée (qui représente par le fait même le temps écoulé dans une expérimentation donnée), du numéro d'expérimentation, ainsi que de 220 autres colonnes représentant chacune l'état d'un capteur filaire.

Tableau 4 – Exemple de données du réseau de capteurs filaires

ID	Numéro d'EXP	Contact 01	Contact 02	Mouvement 01	Débit 01
1634	EXP28	1	1	1	40
1635	EXP28	1	0	0	0
1636	EXP28	1	0	0	0
0	EXP29	0	1	1	20

Cette table contient ainsi entre 2000 et 4000 tuples par expérimentation (selon la durée de celle-ci et le niveau d'activité du participant), totalisant près de 70000 tuples pour 24 expérimentations.

5.3.1.2 Données du réseau de capteurs sans fil

Le système d'acquisition de données du réseau de capteurs sans fil est, quant à lui, conçu pour enregistrer les données en suivant une approche événementielle, c'est-à-dire qu'à chaque événement (par exemple, une ouverture de porte ou une variation dans la consommation électrique), le système reçoit :

- L'information sur le changement d'état du capteur;
- Le temps (à la seconde près) auquel l'évènement est arrivé;
- Le nom du capteur.

Ces informations sont dès lors enregistrées dans la table appropriée (Tableau 5). Cette table contient ainsi entre 150 et 800 tuples par expérimentation (selon la durée de celle-ci et le niveau d'activité du participant), totalisant près de 10000 tuples pour 24 expérimentations.

Tableau 5 – Exemple de données du réseau de capteurs sans fil

Numéro d'EXP	Capteur	Statut	Horodatage
EXP28	Mouvement Salon 01	on	'2015-03-06 11:57:13-05'
EXP28	Mouvement Salon 01	off	'2015-03-06 11:57:23-05'
EXP28	Mouvement Salle de bain 01	on	'2015-03-06 12:00:36-05'
EXP28	Électricité Cuisine 01	712	'2015-03-06 12:00:42-05'
EXP28	Mouvement Salle de bain 01	off	'2015-03-06 12:00:36-05'
EXP28	Électricité Cuisine 01	0	'2015-03-06 12:01:01-05'
EXP28	Contact Entrée 01	on	'2015-03-06 12:12:01-05'
EXP29	Mouvement Entrée 01	on	'2015-03-06 15:37:48-05'

5.3.1.3 Données audiovisuelles

Nous avons recueilli une grande quantité de données audiovisuelles, totalisant près de 400 gigaoctets de données. Quatre fichiers audio au format non compressé **WAV**, un par micro, ainsi que six fichiers vidéo, un par caméra, sont créés pour chaque expérimentation. Les fichiers vidéo sont au format **MP4** pour les caméras IP et **MTS** pour les caméscopes, deux formats assez répandus. Les vidéos, une fois traitées, sont disponibles au visionnement sur l'application CoLab.

5.3.1.4 Données d'annotation de la durée des tâches issues d'APD

Ces données sont initialement stockées dans la BD SQLite interne de la tablette Android utilisée. Après extraction à l'aide du script `sqlite_get.bat` (Annexe C.1), elles seront stockées sur un serveur du laboratoire DOMUS en format **db** et consultées à l'aide du logiciel **SQLiteBrowser** [40]. Le Tableau 6 présente un extrait de la table qui contient les données d'annotation sur la durée des tâches. Le Tableau 7 présente un extrait de la table qui contient les heures de début et de fin de chaque expérimentation, ainsi que son état. Cette information permet de savoir si une expérimentation n'a pas été terminée.

Tableau 6 – Exemple de données d'annotation de la durée des tâches

ID	Numéro d'EXP	ID de tâche	Statut	Horodatage
1	EXP28	1	ACTIVE	2015-03-06_11h48.55
2	EXP28	14	ACTIVE	2015-03-06_11h49.12
3	EXP28	15	ACTIVE	2015-03-06_11h49.42
4	EXP28	14	PAUSED	2015-03-06_11h49.43
5	EXP28	15	FINISHED	2015-03-06_11h51.12
6	EXP28	14	ACTIVE	2015-03-06_11h51.15
7	EXP28	1	FINISHED	2015-03-06_11h53.26

Tableau 7 – Exemple de données d’heures de début et de fin d’expérimentation

Numéro d’EXP	Heure de départ	Heure de fin	Temps écoulé	État
EXP28	2015-03-06_11h22.46	2015-03-06_12h12.19	00h49.33	FINISHED
EXP29	2015-03-06_15h12.39	NULL	NULL	ACTIVE

5.3.1.5 Données d’observation des spécialistes

Il a été question, dans la section 4.2.2, de l’utilisation des observations des spécialistes pour bonifier les données de capteurs. Ces observations sont divisées en deux groupes : erreurs ou assistances. Elles auraient pu être comptabilisées, étant donné que nous savons déjà qu’il y a corrélation entre la quantité d’observations et le déclin cognitif des participants¹¹. Par contre, la comptabilisation et l’exportation de ces données sont des opérations fastidieuses dans la version courante de l’application CoLab, parce que cette fonctionnalité n’était pas prioritaire dans le cadre de la maîtrise de mon collègue Jules Randolph. Par conséquent, l’utilisation de ces observations dans le but de bonifier le jeu de données des capteurs devra faire partie de travaux futurs.

5.3.2 Prétraitement des données

Nous verrons ici le prétraitement nécessaire pour pouvoir utiliser chaque type de données recueillies.

5.3.2.1 Données du réseau de capteurs filaires

Voici la liste des étapes de prétraitement effectuées sur les données des capteurs filaires. Les scripts développés et utilisés sont disponibles à l’Annexe C.

¹¹ Si on se base sur le système de cotation du Profil des AVQ [6].

PF1. Exporter les données en format CSV pour simplifier le prétraitement et l'analyse des données. Les nombres décimaux ayant été stockés dans la BD avec des virgules comme séparateur décimal, il faut une façon de bien exporter les valeurs pour respecter le format CSV, qui sépare chaque valeur par une virgule. Un script **Microsoft Powershell** [41] a été utilisé pour cela.

PF2. Supprimer les données superflues avant et après l'expérimentation.

- Pour déterminer le début, en premier lieu, il fallait identifier le moment où la porte d'entrée s'ouvre, ce qui marque le début de l'expérimentation. Pour cela, nous disposons de l'heure de départ officielle grâce à la tablette Android (avec environ une à deux secondes d'imprécision). Une fois ce moment trouvé, on supprime toutes les données précédant le premier « 0 » associé au capteur de contact de la porte, excepté la toute dernière ligne à 1 (pour bien voir la transition au début de l'expérimentation). Ainsi, on commence **dès que** la porte est détectée comme étant ouverte.
- Pour déterminer la fin, nous avons, grâce à la tablette Android, l'heure de fin et donc la durée de l'expérimentation. Nous n'avons qu'à convertir cette durée en secondes et ajouter cette valeur à l'ID du point de départ établi précédemment pour trouver l'ID de la fin et supprimer toute donnée après ce point. Rappelons que l'ID correspond au nombre de secondes écoulées depuis le démarrage de l'acquisition des données pour une expérimentation en particulier.

PF3. Concaténer les données de toutes les expérimentations en un seul fichier en utilisant le script codé en **Python** [42] décrit à l'Annexe C.2.

PF4. Retoucher les données pour qu'elles soient utilisables avec le logiciel **Weka** [43], pour exécuter des filtres sur les données. Ces retouches incluent : transformer les nombres décimaux à virgules en décimaux à point (Annexe C.3) et supprimer les

espaces superflues dans les chaînes de caractères. À ce point-ci, le fichier de données a une taille sur disque approximative de 70Mo.

PF5. Utiliser le filtre **RemoveUseless** implémenté dans Weka avec le paramètre par défaut (99). Il supprime les attributs qui ne varient jamais, donc les capteurs qui ne sont jamais déclenchés dans aucune expérimentation. Cela réduit le nombre d'attributs de 223 à 120.

PF6. Retirer les attributs représentant les capteurs de température. En effet, nous avons remarqué trop tard que les capteurs de température étaient mal calibrés (les valeurs de température ambiante varient entre 34°C et 41°C), et nous jugeons qu'ils ne fournissent aucune information intéressante. Cela réduit le nombre d'attributs de 120 à 104.

PF7. Retirer divers capteurs analogiques, dont l'accéléromètre sur le lit (détecte une pression sur le lit), l'accéléromètre du bain et les capteurs à infrarouge de la cuisinière, ainsi que le capteur de signal de marche de la cuisinière. Ces capteurs n'étaient pas très utiles, mais surtout mal calibrés. Rappelons que dans ce projet, l'accent avait été mis sur les capteurs sans fil et les capteurs filaires correspondants. Cela réduit le nombre d'attributs de 104 à 95.

PF8. Retirer les capteurs de luminosité. Nous ne croyons pas que ces capteurs nous fourniront une information intéressante. De plus, plusieurs de ces capteurs ne semblent pas avoir bien fonctionné pour plusieurs expérimentations. Cela réduit le nombre d'attributs de 95 à 82.

PF9. Retirer encore quelques capteurs inconnus ou divers, comme des débitmètres qui ne se sont déclenchés qu'à cause d'un reflux d'eau, ainsi que deux colonnes d'information redondante sur l'expérimentation. Cela réduit le nombre d'attributs de 82 à 74.

PF10. Binariser les valeurs des débitmètres en appliquant une fonction de seuil à l'aide du script **standardizeFlowmeter** (Annexe C.4). Comme les valeurs reçues des

débitmètres sont discrètes, entre 0 et 1000 et par pas de 10, toute valeur de débitmètre différente de 0 (qui indique un débit d'eau non nul) sera remplacée par la valeur 1. Nous aurons donc des données simplifiées du type « ouvert/fermé ».

PF11. Finalement, le script **standardizeFlowmeter** (Annexe C.4) s'occupe aussi d'inverser les valeurs binaires des capteurs de contact et de mouvement. En effet, pour ces capteurs, un 0 signifiait « ouvert / mouvement détecté » et un 1 signifiait « fermé / pas de mouvement ». Ces valeurs ont été inversées pour être plus logiques et pour respecter le standard des capteurs sans fil.

Enfin, nous obtenons 74 attributs, incluant le nom de l'expérimentation associé à chaque ligne des données. La taille occupée sur disque par le fichier final est d'environ 11Mo.

5.3.2.2 Données du réseau de capteurs sans fil

Voici maintenant la liste des étapes de prétraitement effectuées sur les données des capteurs sans fil.

PSF1. Exporter les données en format CSV pour simplifier le prétraitement et l'analyse des données. Aucun traitement supplémentaire, comme c'était le cas pour PF1, n'est requis.

PSF2. Supprimer les données superflues avant et après l'expérimentation.

- Pour déterminer le début, en premier lieu, il fallait identifier le moment où la porte d'entrée s'ouvre, ce qui marque le début de l'expérimentation. Pour cela, nous disposons de l'heure de départ officielle grâce à la tablette Android (avec environ une à deux secondes d'imprécision). Une fois ce moment trouvé, on supprime toutes les données précédant le premier « 0 » associé au capteur de contact de la porte. Ainsi, on commence **dès que** la porte est détectée comme étant ouverte.

- Si le capteur de la porte d'entrée n'a pas fonctionné, nous prenons comme point de départ celui marqué par la tablette, en vérifiant avec la vidéo si le tout semble cohérent et synchronisé.
- Pour déterminer la fin, nous avons, grâce à la tablette Android, l'heure de fin et donc la durée de l'expérimentation. Nous n'avons qu'à ajouter cette valeur à l'horodatage du point de départ établi précédemment pour déterminer la fin et supprimer toute donnée après ce point.

PSF3. (optionnel) Cette étape aurait été de binariser les valeurs analogiques, donc celles des capteurs de consommation électrique. Par contre, cela n'a pas été nécessaire dans le cadre de cette maîtrise pour les analyses effectuées sur ces données.

PSF4. Corriger les déclenchements erronés des capteurs de mouvement en regardant toutes les vidéos d'expérimentation et en comparant ce que l'on observe avec les états des capteurs de mouvement dans les données (voir Section 5.2). Chaque déclenchement erroné est noté comme étant *untrue*, au lieu de *true*.

- Si un déclenchement est erroné à un moment donné, mais que, par exemple, le participant passe devant le capteur dix secondes plus tard et le capteur ne se déclenche pas, garder la valeur précédente erronée quand même.
- Ainsi, tolérer plus ou moins dix secondes d'écart entre la vidéo et les capteurs, pour gérer les délais de transmission ou problèmes de synchronisation.

5.3.2.3 Données audiovisuelles

Lors de l'acquisition vidéo à partir des caméras IP, nous utilisons le logiciel VLC pour acquérir le flux vidéo des caméras à l'aide du protocole RTSP [44] et l'enregistrer dans un fichier au format MP4. Dû à un problème de configuration et au fait que le module de VLC utilisé n'était encore qu'en version beta, les fichiers MP4 produits n'étaient lisibles que par VLC, donc non standards ni compatibles avec la plupart des lecteurs vidéo, dont le lecteur HTML5 [45], utilisé

par CoLab. Il nous fallait alors les convertir au format MP4 standard. Nous avons donc conçu le script **Ipcam-convert**¹² pour accélérer le processus de conversion de plus de 150 fichiers. Au cours des expérimentations, nous avons envisagé de passer au logiciel **avconv** pour l'acquisition vidéo, mais cela aurait requis encore beaucoup d'apprentissage et de configuration. Une dernière solution aurait été d'utiliser un logiciel payant, comme ceux utilisés pour les caméras de surveillance, mais ils sont chers. La seule caractéristique nécessaire d'un tel logiciel est la possibilité de faire de l'acquisition vidéo parallèle et synchrone en utilisant RTSP.

De plus, pour les premières expérimentations au laboratoire DOMUS, nous n'avions pas trouvé comment faire pour que les caméras IP captent le son en même temps que la vidéo, et ce, de façon stable. Nos premiers essais étant infructueux, nous préférons donc ne capter que la vidéo de façon stable. Par conséquent, l'acquisition du son était faite séparément, à l'aide des micros installés dans le laboratoire. À la fin d'une expérimentation, nous obtenions donc un fichier vidéo et un fichier audio, qu'il fallait combiner. Cela représente un défi, car pour certaines vidéos, lorsque l'on parvenait à synchroniser la vidéo avec le son dans les premières minutes, ce ne l'était pas pour la fin de la vidéo. C'est comme si l'acquisition de l'une ou l'autre des sources ne s'était pas fait en temps linéaire, peut-être à cause d'un ralentissement matériel. Dans ce cas, la solution était de trouver un juste milieu, comme de synchroniser à la moitié de la vidéo, ce qui laissait qu'un petit décalage au début ou à la fin de la vidéo.

Enfin, nous avons utilisé les logiciels Windows Movie Maker [46] pour ajouter le son, puis Avidemux [47] pour concaténer les vidéos en plusieurs parties qui étaient séparées en plusieurs fichiers (les caméscopes créent un nouveau fichier lorsque le fichier courant dépasse 2 Go).

¹² Le script a été conçu par mon collègue Jules Randolph. Par contre, c'est principalement moi qui me suis chargé de l'utiliser pour convertir les vidéos, ainsi que des étapes suivantes comme ajouter l'audio.

5.4 Conclusion

Nous avons détaillé dans ce chapitre l'ensemble des données recueillies dans le cadre de ce projet, le prétraitement effectué sur les données pour les nettoyer, ainsi que les hypothèses qui appuient les représentations de données utilisées en vue de la prochaine étape qu'est l'analyse de ces données présentée dans le prochain chapitre. Nous avons aussi explicité différents problèmes survenus durant les expérimentations et comment ils ont affecté l'acquisition de données.

Chapitre 6

Analyse

Afin de vérifier nos hypothèses¹³ sur le potentiel des habitats intelligents pour le dépistage précoce de la DTA, il a fallu mettre en place toute une chaîne : 1) effectuer des expérimentations en environnement intelligent, 2) acquérir des données, 3) annoter et prétraiter ces données. Ce chapitre présente le dernier maillon de cette chaîne qui consiste à exploiter la masse d'informations obtenues. La section 6.1 expliquera comment les données sont représentées afin de les analyser. La section 6.2 explorera comment l'apprentissage supervisé peut être utilisé pour prédire la classe cognitive d'un participant. La section 6.3 suivra une autre voie et exploitera des analyses statistiques¹⁴ à cette fin. Finalement, la section 6.4 investiguera comment mener des analyses conjointes entre les deux laboratoires, dans le but de vérifier la faisabilité de réaliser une expérimentation multisite et valider qu'il est possible de générer des données similaires, comparables et pouvant être composées.

6.1 Représentation des données

Une fois les expérimentations terminées, les données récoltées et prétraitées, il reste une dernière question à régler avant l'apprentissage automatique : comment représenter ces données pour servir d'entrée aux divers algorithmes de classification automatique? En effet, les algorithmes de classification prennent en général un seul vecteur par instance d'une classe. Par exemple, l'algorithme C4.5 [48] (implémenté sous le nom de J48 [49] dans le logiciel Weka) prend en entrée un ensemble d'instances (vecteurs), chacune caractérisé par un ensemble d'attributs (colonnes) et étiquetée selon la classe associée à cette instance. Dans notre

¹³ La première hypothèse est qu'une personne atteinte de troubles cognitifs (TCL ou DTA) aura plus de difficulté à se souvenir où vont les objets utilisés que les personnes dites saines. La deuxième hypothèse est qu'une personne atteinte de troubles cognitifs (TCL ou DTA) prendra plus de temps pour accomplir les tâches demandées qu'une personne saine, et donc l'expérimentation au complet.

¹⁴ menées par Maxime Lussier du CRIUGM

cas, une classe représente le diagnostic cognitif d'un participant. Nous avons un participant par expérimentation et une instance par vecteur; il nous faut donc un vecteur par expérimentation. Par contre, nos fichiers de données contiennent actuellement quelques milliers de vecteurs par expérimentation. Il faut alors combiner ces milliers de vecteurs en un seul (voir Figure 23). La solution choisie dépend de l'hypothèse sous-jacente à quelles difficultés cognitives sont observables lors de la réalisation de tâches complexes.

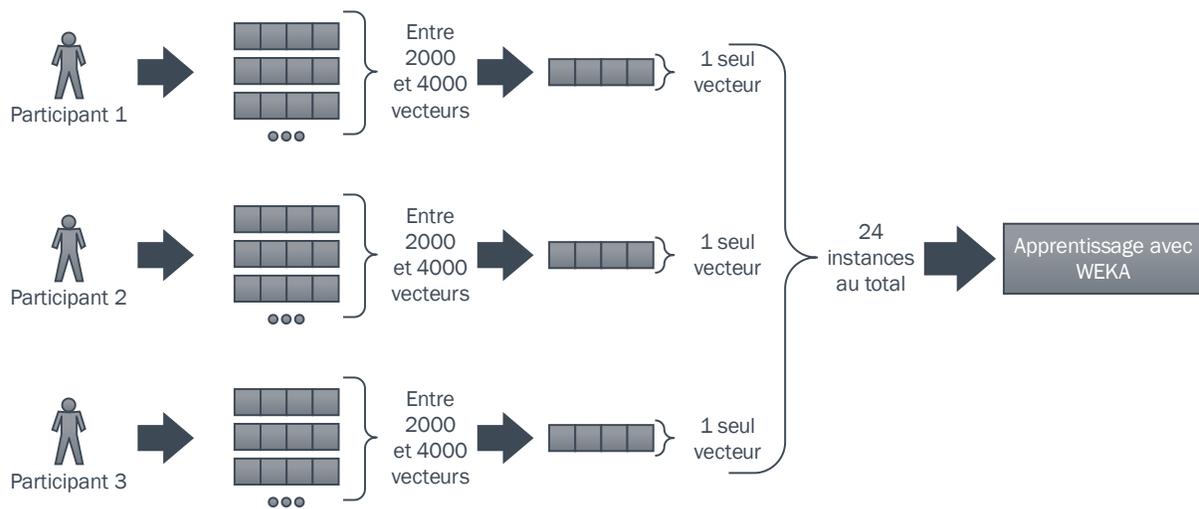


Figure 23 – Technique utilisée pour représenter les données en un seul vecteur par participant

6.1.1 Nombre de déclenchements des capteurs

Comme explicité dans l'introduction, l'hypothèse principale utilisée dans cette recherche est qu'une personne atteinte de troubles cognitifs (TCL ou DTA) aura plus de difficulté à se souvenir où vont les objets utilisés que les personnes dites saines. De plus, leur capacité à planifier l'exécution de tâches complexes et à s'adapter à la nouveauté étant moindre, nous pensons qu'ils chercheront les objets nécessaires de façon possiblement moins efficace. Par exemple, par raisonnement et expérience, une personne saine peut deviner que la poubelle pourrait se trouver sous l'évier, le café et les filtres près de la cafetière, les couverts et ustensiles dans un tiroir coulissant, la vaisselle dans une armoire du haut et les casseroles dans une grande armoire du bas. D'après la littérature (voir section 1.1), une personne présentant des troubles

cognitifs aura probablement moins tendance à réfléchir à une stratégie de recherche et pourrait simplement chercher ces objets un peu n'importe où. Tout cela entraînera une augmentation du nombre de portes d'armoires ouvertes et de mouvements effectués avant d'arriver au but recherché. C'est pourquoi nous pensons que les capteurs de contact sur les portes et les capteurs de mouvement devraient se déclencher significativement plus souvent.

Dans ce cas, la solution utilisée pour représenter les données sera de les convertir en « nombre de déclenchements » de chaque capteur. Chaque colonne représentera un capteur et chaque valeur du vecteur représentera le nombre de fois que ce capteur aura changé d'état au cours de l'expérimentation. Comme toutes les données de capteurs ont été binarisées (voir PF9 à la section 5.3.2.1 précédente), un changement d'état se remarque par une variation de la valeur d'état d'un capteur de zéro à un ou d'un à zéro. Ainsi, ouvrir puis fermer une porte compte pour deux déclenchements et passer devant un capteur de mouvement compte pour deux aussi (passer devant le capteur, puis ne plus être devant). Cette solution nous donnera effectivement un seul vecteur par expérimentation (Tableau 8).

Tableau 8 – Exemple de données représentées en nombre de déclenchements

Numéro d'EXP	Contact 01	Contact 02	Contact 03	Mouvement 01	Débit 01
EXP27	4	6	2	64	6
EXP28	6	2	4	48	4
EXPDTA16	10	8	11	92	10

Cette représentation des données sera utilisée pour l'analyse par classification automatique subséquente. Elle ne regroupe, dans le cadre de cette maîtrise, que les données des capteurs filaires, car elles sont plus fiables et presque sans erreur.

6.1.2 Temps passé dans chaque pièce et durée d'utilisation

Une deuxième hypothèse à vérifier est celle qu'une personne atteinte de troubles cognitifs (TCL ou DTA) prendra plus de temps pour accomplir les tâches demandées qu'une personne saine, et donc l'expérimentation au complet. C'est cette hypothèse que mon collègue Maxime Lussier du CRIUGM a voulu tester. Pour commencer, les données des capteurs sans fil du laboratoire DOMUS et du CRIUGM ont été combinées, tout en gardant en note l'origine des données. Ensuite, le temps passé dans chaque pièce est déduit à partir des capteurs de mouvement (un participant est considéré présent dans la dernière pièce où il a été détecté aussi longtemps qu'il n'a pas été détecté dans une autre pièce). De plus, les temps d'utilisation du four à micro-ondes et de la cafetière sont mesurés. Plus précisément, nous considérons la période où la consommation électrique est supérieure à 5 watts). Finalement, la durée totale d'ouverture de la porte du frigo, d'un tiroir de la chambre (celui où sont rangés les vêtements) et de deux portes d'armoires (contenant la vaisselle nécessaire pour accomplir les tâches demandées) de la cuisine a été mesurée. Plus précisément, nous considérons la période où les deux parties du capteur de contact magnétique étaient séparées, donc tant que la porte est ouverte). Ces données sont cumulées dans un tableau (Tableau 9) pour ensuite faire l'objet d'analyses statistiques. Dans le cadre de cette maîtrise, cette représentation des données ne concerne que les données des capteurs sans fil pour des raisons de cohérence entre les types de capteurs des deux appartements.

Tableau 9 – Exemple de données représentées en temps passé dans la pièce

Numéro d'EXP	Emplacement	Salle de bain	Salon	Salle à manger	Cuisine
EXP27	DOMUS	261	1095	305	1077
EXP45	CRIUGM	122	367	258	1040
EXPDTA16	DOMUS	362	1439	287	1154

6.2 Classification automatique à l'aide des capteurs filaires

La classification automatique sert à catégoriser des objets à l'aide d'algorithmes d'apprentissage automatique. Son objectif est d'attribuer une classe à chaque objet à classer. Dans le cadre du projet DEI, les « objets » à classer sont les participants des expérimentations. Les étapes pour en arriver à une classification des participants sont :

- 1) Des expérimentations sont réalisées lors desquelles les participants effectuent les tâches demandées;
- 2) Les données sont ensuite recueillies, prétraitées et annotées;
- 3) La majeure partie des instances est étiquetée selon le diagnostic (connu d'avance) de chaque participant;
- 4) Cette même partie des instances forme l'ensemble d'apprentissage et est fournie à un algorithme d'apprentissage automatique;
- 5) Le restant des instances compose l'ensemble de test et est ensuite fourni au même algorithme;
- 6) L'algorithme d'apprentissage automatique retourne une prédiction de diagnostic pour chaque instance de l'ensemble de test.

Pour la classification automatique des données issues des capteurs filaires, nous utilisons le logiciel Weka (version 3.8) de l'Université de Waikato en Nouvelle-Zélande qui regroupe plusieurs implémentations en Java de divers algorithmes d'apprentissage automatique et de prétraitement de données. Avant de soumettre nos données à Weka, quelques manipulations sont nécessaires. Premièrement, il faut retirer l'attribut (colonne) qui identifie le nom de l'expérimentation, car un algorithme pourrait l'utiliser pour apprendre et déduire, par exemple, que toutes les expérimentations ayant « DTA » dans leur nom appartiennent à la classe DTA, ce qui causerait un biais substantiel. Deuxièmement, il faut ajouter la classe associée à chaque instance, donc à chaque expérimentation. On ajoute les mots « Sain », « TCL » ou « DTA », le cas échéant, dans la colonne de l'attribut « Classe » ajouté. Finalement, il est impératif de mélanger les instances pour maximiser les chances d'avoir un ensemble d'entraînement

représentatif de toutes les classes. Pour ce faire, nous avons utilisé le filtre **Randomize** de Weka avec la valeur par défaut (42) comme graine (*seed*). Rappelons que les données utilisées sont composées de 24 instances (une par expérimentation : 10 Sain, 10 TCL et 4 DTA) et de 74 attributs.

Dans cette section, nous verrons : les algorithmes d'apprentissage supervisé utilisés, les premiers résultats de classification et leur analyse, la formation d'un sous-ensemble d'attributs les plus significatifs et les résultats obtenus avec ce sous-ensemble d'attributs, les résultats d'apprentissage sur deux classes seulement puis finalement la classification avec fenêtres temporelles.

6.2.1 Algorithmes utilisés

Dans le cadre des analyses effectuées au DOMUS, six algorithmes de classification automatique implémentés dans Weka ont été utilisés : J48, RandomForest, NaiveBayes, OneR, SVM et ZeroR. Chaque algorithme sera décrit et de quelle façon ils ont été utilisés. Les paramètres par défaut de Weka ont été utilisés pour simplifier l'analyse, mais aussi dans le but d'identifier un algorithme efficace sans avoir à passer trop de temps à affiner les paramètres pour chaque site d'expérimentation. De plus, trop affiner les paramètres peut mener au surapprentissage, ce qui nuirait à la capacité de généralisation du modèle sur d'autres sites. Enfin, cette analyse se veut être un premier tour d'horizon des algorithmes existants.

À noter que pour tous les algorithmes et pour toutes les analyses effectuées au DOMUS, nous avons utilisé une technique de validation croisée dix fois (*10-fold cross-validation*) [50] dans le but de valider le modèle obtenu par l'exécution de chacun des algorithmes utilisés. Cette technique sert à vérifier à quel point le modèle est capable de généraliser à un ensemble de données indépendant. L'idée générale est de prendre un petit sous-ensemble de données comme données de test (10% des données pour une validation croisée dix fois) et le reste comme données d'apprentissage. On répète l'opération avec des sous-ensembles différents un nombre de fois prédéterminé, dix fois dans notre cas (quatre pour l'exemple présenté à la Figure 24).

Dix est la valeur par défaut dans Weka, mais nous avons fait quelques tests avec une valeur différente sans variation significative.

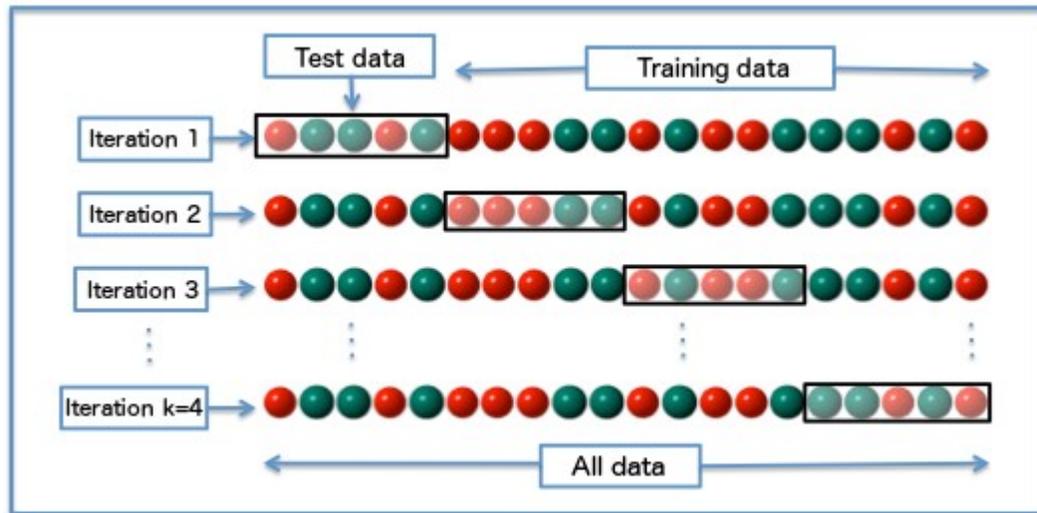


Figure 24 – Fonctionnement de la validation croisée (source : [50])

Une des raisons principales pour utiliser la validation croisée au lieu d'autres techniques de validation conventionnelles¹⁵ est le manque d'instances à classer. En effet, vu que nous n'avons que 24 instances (participants) pour les trois classes, il est difficile de séparer les données en ensembles d'apprentissage et de test sans avoir une perte significative de la capacité de modélisation ou de test. Enfin, la validation croisée fait la moyenne des moyennes des mesures d'ajustement (erreur de prédiction) pour en arriver à une estimation plus précise de la capacité de modélisation de l'algorithme utilisé.

6.2.1.1 C4.5 (implémenté sous le nom de J48)

C4.5 [48] est un algorithme utilisé pour générer des arbres de décision développé par Ross Quinlan. Il est implémenté en Java sous le nom de J48 dans Weka. Les arbres de décision générés par C4.5 peuvent être utilisés pour faire de la classification. Les auteurs du logiciel Weka considèrent que C4.5 est un algorithme d'arbre de décision de référence. Il est

¹⁵ Par exemple partitionner les données en deux ensembles de 70% pour l'apprentissage et 30% pour les tests.

probablement le plus utilisé en pratique jusqu'à maintenant [43]. De plus, C4.5 est devenu assez populaire après avoir été classé #1 dans le prééminent article *Top 10 Algorithms in Data Mining* en 2008 [51]. C'est donc sans surprise que nous avons voulu l'essayer sur nos données.

Voici comment C4.5 fonctionne. Pour chaque nœud de l'arbre de décision à créer, C4.5 choisit l'attribut parmi les données qui divise le plus efficacement possible son ensemble d'instances en sous-ensembles séparant le mieux les classes. Le critère de division utilisé est le gain d'information (la différence d'entropie). L'attribut ayant le plus grand gain d'information est choisi pour faire la décision. On répète pour chaque sous-ensemble jusqu'à ce qu'il ne reste qu'une seule classe ou qu'un nombre minimal d'instances soit atteint (2 instances par nœud par défaut). Ensuite, on procède à l'élagage de l'arbre en retirant les feuilles qui n'apportent aucune amélioration de la précision.

6.2.1.2 RandomForest

RandomForest [52] est l'algorithme qui a offert la meilleure précision de classification dans les travaux pour la mise en place d'un système de positionnement à l'intérieur d'environnements intelligents basé sur les arbres de décision et la technologie RFID passive [53]. Ces travaux ont été menés au DOMUS par Frédéric Bergeron, étudiant au doctorat en informatique. En plus d'offrir les meilleurs résultats dans ce cas précis, RandomForest s'avère plus rapide à entraîner et à utiliser que plusieurs de ses concurrents, dont NaiveBayes.

RandomForest consiste à utiliser 100 Random Trees (arbres de décision aléatoire) et les entraîner séparément. Ces Random Trees sont des arbres de décision où seulement un sous-ensemble d'attributs (choisi aléatoirement) est considéré pour chaque division. Par conséquent, ils sont bien adaptés pour les jeux de données où l'on retrouve un grand nombre d'attributs. Enfin, la classe prédite par RandomForest est alors simplement la classe modale¹⁶ parmi les 100 classes prédites par les Random Trees. Comme nous avons un nombre considérable

¹⁶ La classe ayant la plus grande fréquence.

d'attributs, nous voulions tester si cet algorithme avait des résultats aussi probants que ceux obtenus au DOMUS dans un autre contexte.

6.2.1.3 NaiveBayes

NaiveBayes [54], ou classificateur bayésien naïf, est un classificateur probabiliste simple basé sur le théorème de Bayes [55] où on suppose une forte indépendance (dite naïve) des hypothèses. Malgré cette supposition « naïve » et simpliste, les classificateurs bayésiens naïfs ont fait preuve d'une efficacité surprenante dans plusieurs cas complexes, ce qui leur vaut la neuvième place dans le *Top 10 Algorithms in Data Mining* en 2008 [51]. Le principal avantage du classificateur bayésien naïf est qu'il nécessite relativement peu de données d'entraînement pour estimer les paramètres nécessaires à la classification. Ces paramètres sont les moyennes et variances des différentes variables. Le classificateur bayésien naïf est aussi simple à utiliser et ne demande aucune estimation de paramètres.

Le théorème de Bayes (Figure 25) permet de calculer la probabilité a posteriori, $P(c|x)$, à partir de :

- la probabilité a priori de la classe c , $P(c)$
- la probabilité a priori des attributs $P(X)$ et de
- la vraisemblance (*likelihood* en anglais) $P(X|c)$.

$$P(c|X) = \frac{P(X|c) P(c)}{P(X)}$$

Figure 25 – Théorème de Bayes

Pour calculer la probabilité a posteriori, il faut commencer par construire une table de fréquences pour chaque attribut versus chaque classe. Ensuite, on transforme cette table de fréquences en table de vraisemblances (table de probabilités), puis on utilise l'équation de Bayes (Figure 26) pour calculer la probabilité a posteriori de chaque classe. La classe ayant la plus grande probabilité a posteriori sera la classe prédite par le modèle.

$$P(c|X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

Figure 26 – Équation de Bayes utilisée par NaiveBayes

6.2.1.4 OneR

OneR [56] est un algorithme de classification simple qui donne d'assez bons résultats sur des jeux de données communs. Le modèle obtenu classe les données à l'aide d'une règle basée sur un seul attribut. Selon l'étude [56], sur la plupart des jeux de données étudiés, les meilleures de ces règles simples obtenaient une précision de classification comparable aux règles induites par la majorité des autres algorithmes testés. Un classificateur OneR utilise l'attribut qui donne une erreur minimale pour la prédiction, tout en discrétisant les attributs numériques.

6.2.1.5 Sequential Minimal Optimization (SMO)

SMO [57] est un algorithme d'optimisation largement utilisé pour entraîner une machine à vecteur de support (SVM, *Support Vector Machine*) [58]. SVM arrive troisième dans le *Top 10 Algorithms in Data Mining* en 2008 [51], étant considéré comme « à essayer », offrant une des méthodes les plus robustes et précises parmi les algorithmes bien connus. De plus, SVM ne requiert que peu d'instances (une douzaine selon [51]) pour l'entraînement, ce qui est parfait pour nous, et est insensible au nombre de dimensions (il est donc utile lorsqu'on a un grand nombre de dimensions). Les problèmes multi-classes, comme dans le cas présent, sont résolus en classifiant par paires de classes (donc un contre un).

Nous ne détaillerons pas plus le fonctionnement de SVM et SMO, car nous ne nous sommes finalement pas trop attardés à leur fonctionnement et à leur paramétrage, ni aux résultats obtenus grâce à ces algorithmes dans le cadre de ces travaux. Nous nous sommes plutôt concentrés sur OneR et les arbres de décision. Cette famille d'algorithmes pourrait s'avérer prometteuse pour de futurs travaux en considérant une grande quantité de capteurs.

6.2.1.6 ZeroR (probabilités a priori)

ZeroR sert de base pour la classification en trouvant la classe ayant la plus grande fréquence et en classant toutes les instances selon cette classe. Cela revient à trouver les probabilités a priori

de chaque classe. Si nous avons trois classes également distribuées, la précision de classification sera de 33.3%. Par le fait même, la précision augmente à 50% pour un jeu de données à deux classes également distribuées. Les autres algorithmes de classification utilisés doivent obtenir de meilleurs résultats que ZeroR, sinon ils seront jugés inutiles.

6.2.2 Premiers résultats

Les résultats suivants (Tableau 10) représentent une première exploration des données avec divers algorithmes connus et servent de point de départ pour valider nos hypothèses et notre méthodologie. Les taux d'instances correctement classées sont arrondis à l'entier près et l'erreur absolue moyenne à quatre décimales près.

Parmi ces résultats, on peut remarquer que le meilleur taux d'instances correctement classées (la précision de classification) et la plus faible erreur absolue moyenne sont obtenus avec l'algorithme OneR, soit 67% et 0,2222 respectivement. OneR est un algorithme très simple qui n'utilise qu'un seul attribut (un capteur) pour créer la règle suivante:

Nombre de déclenchements du capteur MV1:

< 99.0 → sain
< 155.0 → TCL
>= 155.0 → DTA

Donc, si le nombre de déclenchements du capteur MV1 est inférieur à 99, le participant est classé comme étant sain. Si le nombre est supérieur ou égal à 99, mais inférieur à 155, on classe le participant comme étant TCL. Finalement, si le nombre est supérieur ou égal à 155, le participant sera considéré comme DTA. Ce fameux capteur est le capteur de mouvement de la hotte, au-dessus de la cuisinière (Figure 27), qui surveille la surface de cuisson de la cuisinière.

En somme, ces résultats révèlent qu'avec la représentation des données, les algorithmes et les paramètres utilisés, il est plus efficace de discriminer les participants en n'utilisant qu'un seul capteur que l'ensemble complet de 73 capteurs. C'est un résultat assez surprenant et nous y reviendrons à la section 0. Tout d'abord, nous nous attarderons dans la prochaine section à

vérifier la fiabilité des résultats obtenus avec OneR et de connaître en quoi le capteur MV1 est discriminant. Finalement, on remarque sur l'ensemble des analyses que les participants DTA sont en général très mal classés : au mieux un participant sur quatre a été correctement classé par OneR et SMO (voir les 0 et les 1 au centre des matrices de confusion du Tableau 10). Nous y reviendrons à la section 6.2.5.

Tableau 10 – Premiers résultats de l'apprentissage supervisé avec 3 classes (Sain, DTA et TCL) en utilisant tous les capteurs

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	42%	a b c 7 1 2 a = Sain 2 0 2 b = DTA 4 3 3 c = TCL	0.3875
NaiveBayes	46%	5 1 4 a = Sain 2 0 2 b = DTA 4 0 6 c = TCL	0.36
RandomForest	54%	6 0 4 a = Sain 2 0 2 b = DTA 2 1 7 c = TCL	0.3908
OneR	67%	7 0 3 a = Sain 1 1 2 b = DTA 2 0 8 c = TCL	0.2222
SMO	46%	5 1 4 a = Sain 1 1 2 b = DTA 3 2 5 c = TCL	0.3796
ZeroR	42%	10 0 0 a = Sain 4 0 0 b = DTA 10 0 0 c = TCL	0.4222



Figure 27 – Emplacement du capteur MV1

6.2.3 Fiabilité des résultats avec OneR

Plusieurs questions ont été soulevées sur la fiabilité de la règle créée par l'algorithme OneR : est-ce que le capteur ne s'active que lorsqu'il y a du mouvement (d'un humain) ou dans d'autres conditions? Est-ce que l'endroit où le participant plaçait la feuille d'instructions avait un impact? C'est pourquoi les facteurs qui influencent le déclenchement du capteur MV1 ont été vérifiés lors d'une mini-expérimentation. Le Tableau 11 présente nos questionnements principaux sur la sensibilité du capteur MV1 et les conditions du test pour vérifier son bon fonctionnement.

Suite à ces analyses, nous pouvons conclure que le capteur est « bien placé » : il ne détecte que les mouvements au-dessus de la surface de cuisson. Par contre, cela inclut les mouvements de chaleur causés par de l'eau qui bout dans une casserole ou même parfois par la cuisson des œufs dans une poêle. De plus, la cuisson de l'œuf dans une casserole amène un biais majeur, car cette méthode de cuisson stimulera le capteur plus souvent qu'une cuisson à la poêle, dû au dégagement de chaleur intense, et l'algorithme OneR aura donc tendance à classer le participant comme étant DTA parce qu'il a choisi de cuire ses œufs de cette façon. Rappelons que tous les participants étaient libres de choisir la méthode de cuisson qu'ils voulaient.

Tableau 11 – Tests pour connaître dans quels cas le capteur MV1 se déclenche

Question à répondre (est-ce que le capteur se déclenche...)	Conditions du test de vérification	Réponse
Si une personne passe dans la cuisine pour aller dans la salle de bain?	Cuisinière fermée. 4 passages consécutifs de plus en plus près de la cuisinière	NON
Quand la casserole chauffe?	Casserole sur rond avant-droit, rond avant-gauche chauffe aussi	OUI
Quand l'eau bout?	Casserole sur rond avant-gauche, le plus près possible du capteur, mais 100% sur le rond	OUI
Quand une poêle chauffe sur un rond?	Poêle vide située sur le rond avant-gauche, de taille équivalente au rond	Parfois
Quand le ventilateur de la hotte fonctionne?	Le test précédent avec la poêle avait généré beaucoup de fumée, nous avons donc démarré le ventilateur.	Parfois
Quand la personne consulte la feuille de tâches (qui serait situé sur le comptoir à droite ou à gauche de la cuisinière)?	J'ai fait exprès pour passer très près de la cuisinière.	NON, on doit vraiment passer au-dessus de la surface de la cuisinière pour être vu, donc il est fort peu probable que la position de la feuille de tâches influence le déclenchement de capteurs

L'algorithme OneR semble mieux classer les personnes TCL et DTA que les autres algorithmes. Une des hypothèses possibles pour l'expliquer est qu'un participant présentant des troubles cognitifs prendrait en général plus de temps pour réaliser la tâche de la cuisson de l'œuf qu'un participant sans trouble cognitif. Cependant, il se peut que cette « lenteur » ne soit

pas imputable à la tâche elle-même, mais plutôt au fait que le participant ait un ensemble de tâches à réaliser : les personnes saines organiseraient mieux leur temps, leurs déplacements et l'enchaînement des tâches. Alors, s'il n'y avait que la préparation de repas, on ne verrait probablement pas de différence. Des travaux futurs pourraient vérifier cette hypothèse en demandant aux participants de n'exécuter que la tâche de la préparation de repas.

6.2.4 Capteurs les plus discriminants

On peut retenir de l'exercice précédent que le nombre de capteurs est trop grand et que nos algorithmes ne parviennent pas à tirer profit de cette grande quantité d'information. Une approche alternative serait de créer un ensemble réduit de capteurs, composé des capteurs les plus discriminants, en ignorant tous les autres. Pour ce faire, il faut en premier lieu déterminer quels capteurs sont les plus pertinents. Ensuite, il faut créer un sous-ensemble d'attributs avec ces capteurs seulement. Cette section détaillera l'approche utilisée.

6.2.4.1 Découverte des capteurs les plus discriminants

Pour découvrir les capteurs les plus discriminants, une technique toute simple a été utilisée. Nous voulions savoir quel capteur l'algorithme OneR utiliserait si le capteur le plus discriminant selon lui (MV1) était retiré des données. Dans ce cas, OneR trouverait un autre capteur avec lequel discriminer les données, qui sera considéré comme étant le deuxième capteur le plus discriminant. En répétant cette procédure plusieurs fois, nous obtiendrons une liste des meilleurs capteurs à considérer.

Réduire le nombre d'attributs en ne gardant que les plus discriminants s'appelle la sélection des fonctionnalités [59]. Des techniques plus poussées, appropriées aux algorithmes de classification utilisés, pourront être utilisées lors de travaux futurs. Notre intérêt principal est d'explorer l'analyse de données récoltées dans le cadre du projet et mieux interpréter les résultats étonnants obtenus avec OneR.

6.2.4.2 Analyse avec un ensemble d'attributs réduit

Nous avons donc répété les analyses avec les deux capteurs les plus discriminants, puis les trois capteurs les plus discriminants, ainsi de suite jusqu'aux dix capteurs les plus discriminants (Tableau 12). Le sous-ensemble donnant les résultats les plus significatifs est celui composé des cinq capteurs les plus discriminants:

- CA1, porte d'armoire à gauche de la cuisinière (contient sucre, café)
- CA4, porte d'armoire à droite de la cuisinière (contient assiettes)
- MV1, capteur de mouvement sous la hotte
- MV4, capteur de mouvement couvrant toute la cuisine
- MV5, capteur de mouvement couvrant tout le salon

On peut remarquer, en premier lieu, une amélioration générale des taux d'instances correctement classées (sauf pour RandomForest qui ne varie pas), atteignant le sommet de 71% avec NaiveBayes. Celui-ci a eu de loin la plus grande amélioration (de 46% à 71%) et porte à croire que cet algorithme performe typiquement mieux avec un nombre restreint d'attributs. Les résultats pour OneR restent évidemment les mêmes étant donné qu'il utilise toujours le capteur MV1 pour discriminer.

Le choix d'utiliser cinq capteurs a été justifié par les analyses faites pour un nombre variable d'attributs d'un à dix. Comme la performance des algorithmes ne faisait que diminuer à mesure que le nombre de capteurs augmentait (dès six capteurs), les essais ont été arrêtés à dix capteurs. Les autres capteurs risquent donc d'être soit inutiles, soit de dégrader la qualité de la classification.

De façon générale, il est coutume de supprimer les attributs inutiles, car ceux-ci peuvent dégrader la qualité des modèles entraînés, pour les raisons suivantes [59] :

- Certains attributs sont parasites ou redondants, ce qui rend plus difficile la découverte de séquences significatives à partir des données;

- Pour découvrir des séquences de qualité, la plupart des algorithmes d'exploration de données nécessitent un jeu de données d'apprentissage beaucoup plus important dans le cas d'un jeu de données de grande dimension (avec un grand nombre d'attributs).

Comme nous avons plus d'attributs (73) que d'instances (24), plusieurs algorithmes en souffrent. Cela explique pourquoi la performance de la plupart des algorithmes augmente lorsqu'on n'utilise que les cinq attributs les plus discriminants. C'est pourquoi il faut d'abord caractériser les données et voir ensuite comment appliquer les algorithmes.

Tableau 12 – Résultats de l'apprentissage supervisé avec 3 classes (Sain, DTA, TCL) en utilisant un ensemble réduit d'attributs

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	54%	a b c 7 0 3 a = Sain 1 0 3 b = DTA 4 0 6 c = TCL	0.3265
NaiveBayes	71%	7 0 3 a = Sain 1 2 1 b = DTA 1 1 8 c = TCL	0.2135
RandomForest	54%	7 0 3 a = Sain 1 0 3 b = DTA 3 1 6 c = TCL	0.3364
OneR	67%	7 0 3 a = Sain 1 1 2 b = DTA 2 0 8 c = TCL	0.2222
SMO	63%	7 0 3 a = Sain 1 0 3 b = DTA 2 0 8 c = TCL	0.3426
ZeroR	42%	10 0 0 a = Sain 4 0 0 b = DTA 10 0 0 c = TCL	0.4222

6.2.5 Analyse à deux classes : sain ou avec troubles cognitifs

Un point important de l'interprétation des résultats faite à la section 6.2.2 est que les participants DTA sont en général très mal classés : au mieux un participant sur quatre a été correctement classé. Cela s'explique en partie par le fait qu'il n'y avait que quatre participants et qu'il est plus difficile pour les algorithmes d'apprendre et de généraliser à partir de si peu d'instances. Toutefois, nous voulons pousser l'analyse un peu plus loin et voir si l'on peut obtenir de meilleurs résultats en se restreignant à deux classes seulement. Pour ce faire, deux choix s'offrent à nous :

- Combiner les données des classes TCL et DTA en créant une classe Avec_TC¹⁷, donc faire une comparaison entre les personnes qui ont des troubles cognitifs (classe Avec_TC) et ceux qui n'en ont pas (classe Sain);
- Retirer complètement les participants DTA, pour avoir un jeu de données composé de deux classes également distribuées (10 Sain et 10 TCL).

Nous avons testé les deux approches, étant donné qu'elles sont toutes deux intéressantes et rapides à tester. Nous avons aussi pris le temps de tester ces deux approches avec l'ensemble des capteurs et avec les cinq capteurs les plus discriminants.

6.2.5.1 Utilisation de tous les capteurs pour discriminer entre les 2 classes Sain et Avec_TC

Les résultats présentés dans le Tableau 13 ont été obtenus avec les données de tous les capteurs, en combinant les classes TCL et DTA pour créer la classe Avec_TC.

Si l'on compare aux résultats à trois classes présentés à la section 6.2.2 précédente (Tableau 10), on voit une nette augmentation des taux d'instances correctement classées pour certains algorithmes, atteignant les 75% avec RandomForest et SMO. Par contre, il faut aussi noter le résultat obtenu par ZeroR (probabilités a priori), dont plusieurs algorithmes se servent dans

¹⁷ avec troubles cognitifs

leurs calculs (NaiveBayes, J48 et RandomForest dans notre cas). Une hausse de la fréquence relative de la classe la plus fréquente fera augmenter la performance de classification des algorithmes qui se basent dessus. Donc, une partie de l'augmentation des performances est due au simple fait que nous avons deux classes au lieu de trois.

Tableau 13 – Résultats de l'apprentissage supervisé sur les classes Sain et Avec_TC en utilisant tous les capteurs

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	67%	a b 5 5 a = Sain 3 11 b = Avec_TC	0.3403
NaiveBayes	58%	5 5 a = Sain 5 9 b = Avec_TC	0.4124
RandomForest	75%	6 4 a = Sain 2 12 b = Avec_TC	0.4346
OneR	67%	4 6 a = Sain 2 12 b = Avec_TC	0.3333
SMO	75%	7 3 a = Sain 3 11 b = Avec_TC	0.25
ZeroR	58%	0 10 a = Sain 0 14 b = Avec_TC	0.4891

Nous avons ensuite testé le cas où on retirerait toutes les données des participants DTA, donc on ne garde que les dix participants Sain et les dix TCL (Tableau 14).

Les résultats sont légèrement inférieurs au test précédent, mais rien de significatif. Il se peut donc que les données des quatre participants DTA ne soient pas assez différentes des TCL pour différencier ces deux classes, et que les supprimer totalement prive les algorithmes de quatre instances utiles à l'apprentissage.

Tableau 14 – Résultats de l'apprentissage supervisé avec 2 classes (Sain et TCL)
en utilisant tous les capteurs

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	50%	a b 6 4 a = Sain 6 4 b = TCL	0.5
NaiveBayes	65%	6 4 a = Sain 3 7 b = TCL	0.3311
RandomForest	65%	7 3 a = Sain 4 6 b = TCL	0.4285
OneR	55%	5 5 a = Sain 4 6 b = TCL	0.45
SMO	70%	7 3 a = Sain 3 7 b = TCL	0.3
ZeroR	50%	10 0 a = Sain 10 0 b = TCL	0.5

6.2.5.2 Utilisation d'un sous-ensemble de capteurs les plus discriminants entre les deux classes Sain et Avec_TC

Ayant déjà eu de bons résultats en n'utilisant que les cinq capteurs les plus discriminants, à la section 0, nous avons répété les essais décrits en 6.2.5.1 avec les capteurs les plus discriminants. Comme précédemment, les participants TCL et DTA sont combinés pour former la classe Avec_TC (Tableau 15).

Ce sont les meilleurs résultats obtenus jusqu'à présent, atteignant les 79% avec l'algorithme NaiveBayes, qui est passé de 58% à 79%. Par contre, la performance des autres algorithmes n'est presque pas supérieure aux résultats du Tableau 13 (Sain et Avec_TC, mais 73 attributs, soit tous les capteurs).

Tableau 15 – Résultats de l'apprentissage supervisé avec les classes Sain et Avec_TC en utilisant le sous-ensemble des cinq capteurs les plus discriminants

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	71%	a b 5 5 a = Sain 2 12 b = Avec_TC	0.3208
NaiveBayes	79%	7 3 a = Sain 2 12 b = Avec_TC	0.2039
RandomForest	75%	7 3 a = Sain 3 11 b = Avec_TC	0.3754
OneR	71%	5 5 a = Sain 2 12 b = Avec_TC	0.2917
SMO	75%	7 3 a = Sain 3 11 b = Avec_TC	0.25
ZeroR	58%	0 10 a = Sain 0 14 b = Avec_TC	0.4891

Nous avons ensuite testé le cas où on retirait toutes les données des participants DTA, donc on ne garde que les dix participants Sain et les dix TCL (Tableau 16), comme précédemment.

Ces résultats représentent une amélioration intéressante de la performance de classification des algorithmes si l'on compare aux résultats du Tableau 14. Par contre, la moyenne des performances est légèrement inférieure à celle du Tableau 15. Finalement, nous obtenons 80% de performance avec NaiveBayes, ce qui est nettement supérieur à ZeroR.

Tableau 16 – Résultats de l'apprentissage supervisé avec les classes Sain et TCL en utilisant le sous-ensemble des cinq capteurs les plus discriminants

Algorithme utilisé	Taux d'instances correctement classées	Matrice de confusion	Erreur absolue moyenne
J48	60%	a b 5 5 a = Sain 3 7 b = TCL	0.3909
NaiveBayes	80%	7 3 a = Sain 1 9 b = TCL	0.2091
RandomForest	70%	7 3 a = Sain 3 7 b = TCL	0.3495
OneR	55%	5 5 a = Sain 4 6 b = TCL	0.45
SMO	75%	7 3 a = Sain 2 8 b = TCL	0.25
ZeroR	50%	10 0 a = Sain 10 0 b = TCL	0.5

6.2.5.3 Conclusion

En somme, les analyses à deux classes omettant les DTA ou les combinant avec les TCL donnent généralement de meilleurs résultats que les analyses à trois classes. Plusieurs explications sont possibles :

- Cela pourrait être attribué au fait que les participants DTA ont une grande variance dans le temps requis pour compléter les tâches et dans le nombre de déclenchements : deux participants ont complété les tâches à la hâte (32m39 et 48m47) et deux autres ont pris beaucoup plus de temps (1h13m55 et 1h20m58). Cela complexifie donc la tâche de trouver des points communs entre ces quatre participants qui permettent de les différencier des autres classes (Sain et TCL), surtout en utilisant une approche basée sur le nombre de déclenchements de capteurs comme nous le faisons.

- Il se pourrait aussi qu'il y ait ou non des différences marquées entre les TCL et les DTA selon la représentation des données utilisée. Ici, utiliser le nombre de déclenchements de capteurs acquis lors d'expérimentations de ce type ne permettrait pas de différencier les TCL des DTA, mais permettrait tout de même de les dissocier des participants sains.
- Finalement, nous n'avons que quatre participants, ce qui est très peu pour former un groupe homogène permettant de généraliser.

6.2.6 Classification avec fenêtres temporelles

Les résultats de notre première exploration des algorithmes d'apprentissage supervisé nous ont amené à considérer d'autres techniques pour améliorer la classification. Les algorithmes de classification avec fenêtres temporelles sont des candidats tout désignés. L'utilisation de fenêtres temporelles est très prometteuse et permet d'avoir beaucoup plus d'instances de classes pour l'apprentissage, quelques milliers au lieu de seulement 24. Ces travaux sont menés par notre collègue Belkacem Chikhaoui du Centre de Recherche Informatique de Montréal (CRIM). Nous avons quelques résultats préliminaires sur les données du réseau de capteurs filaires.

L'algorithme utilisé s'appuie sur une fenêtre temporelle et extrait le nombre de déclenchements de capteurs dans cette fenêtre en utilisant des chevauchements de 50%. Par exemple, si la taille de la fenêtre est de 60 secondes, la première fenêtre comprend les données de 0s seconde à 59s, puis la seconde de 30s à 89s, etc. Les résultats sont encourageants : les meilleurs résultats pour l'instant sont 76.1% de reconnaissance pour les trois classes Sain, TCL et DTA.

6.2.7 Conclusion

En résumé, nos travaux d'analyse préliminaires montrent qu'il est possible de classer les participants selon l'observation de leurs comportements à partir des données recueillies par un environnement intelligent. Premièrement, pour tous les algorithmes utilisés, nous obtenons des résultats meilleurs que la probabilité a priori de la classe la plus fréquente (42%).

Deuxièmement, en utilisant un algorithme simple comme OneR, nous obtenons 67% de taux d'instances correctement classées, ce qui est déjà un bon début. Troisièmement, en combinant les données des participants DTA et TCL, et en n'utilisant comme attributs que les cinq capteurs les plus discriminants, nous obtenons de meilleurs résultats : 80% avec NaiveBayes. Toutefois, il s'avère très difficile de discriminer les participants DTA avec nos données, de par leur faible nombre et la grande variance (dans le temps requis pour accomplir les tâches et dans le nombre de déclenchements de capteurs) entre les participants. Il semble que des techniques plus poussées, comme l'utilisation de fenêtres temporelles, permettraient d'avoir plus d'instances utiles à l'apprentissage, ce qui pourrait aider à discriminer les participants DTA.

6.3 Analyse statistique à l'aide des capteurs sans fil

Cette section traitera des analyses menées par Maxime Lussier du CRIUGM sur les données des capteurs sans fil du laboratoire DOMUS et du CRIUGM. Comme nous avons eu peu de participants atteints d'une DTA, comparativement aux personnes saines et celles atteintes d'un TCL, et que les données des participants DTA ont une très grande variance, nous avons décidé de retirer ces données pour les analyses statistiques.

Le temps total requis pour accomplir les cinq tâches était significativement plus long pour les participants TCL (43.64min en moyenne) que pour les participants sains (34.68min). Par contre, la différence dans la durée peut être presque entièrement attribuée à un plus grand temps passé dans la cuisine chez les participants TCL (28.71min), comparée aux participants sains (22.70min). De plus, les participants TCL ont passé plus de temps à regarder à l'intérieur du frigo (1.32min) que les participants sains (0.98min). Ils ont aussi passé plus de temps à regarder dans les armoires de la cuisine (celles contenant les tasses, les assiettes, le sucre, le café et les filtres) (4.47min) que les participants sains (3.17min). Pour une explication plus détaillée de ces résultats, et d'autres analyses incluant entre autres les tests cognitifs détaillés à la section 2.1.1, voir l'article de Maxime Lussier [60].

Grâce à ces résultats, nous pouvons maintenant affirmer que, en moyenne, les personnes atteintes de troubles cognitifs prennent plus de temps que les personnes saines pour accomplir la plupart des tâches demandées et l'expérimentation au complet.

6.4 Faisabilité d'une expérimentation multisite

Comme les données proviennent de deux centres de recherche différents, une validation interlieu a été faite. Les résultats montrent que les deux lieux sont comparables en termes de temps passé dans chaque pièce, ainsi qu'en utilisation des appareils de la cuisine et du rangement [60]. La seule différence significative observée est que les participants évalués à Montréal passaient plus de temps dans le salon, par exemple à chercher l'horaire d'autobus, indépendamment des capacités cognitives (sain, TCL ou DTA). Cela répond à l'objectif O2¹⁸.

Pour ce qui est du nombre optimal de capteurs pour obtenir des résultats significatifs, nous avons exploré le problème avec nos analyses sur un sous-ensemble des données du DOMUS, en n'utilisant que cinq capteurs. Comme nous avons obtenu de meilleurs résultats de classification des diagnostics cognitifs avec ce sous-ensemble qu'avec un grand nombre de capteurs, cela porte à croire qu'un petit ensemble de capteurs déployé dans l'environnement pourrait être suffisant. Par contre, connaître à l'avance les cinq capteurs les plus discriminants qu'il faudrait placer dans l'environnement s'avère très difficile. D'autres travaux devront être faits pour trouver la réponse, mais l'idée générale est d'avoir suffisamment de capteurs reliés aux tâches les plus discriminantes, sans nécessairement viser le nombre minimal de capteurs. Par exemple, il s'agit de placer des capteurs de contacts sur les portes des armoires qui doivent être ouvertes pour accomplir la tâche de préparation de repas et peut-être les portes à proximité. Évidemment, un plus petit ensemble de capteurs serait plus facile à déployer sur une multitude

¹⁸ Vérifier la faisabilité de réaliser une expérimentation multisite, en impliquant deux laboratoires intelligents et en vérifiant la possibilité de générer des données similaires, comparables et pouvant être composées.

de sites dans le futur, demandant moins de temps et d'argent pour l'installation. Cela répond à l'objectif O3¹⁹.

¹⁹ Déterminer le type optimal de réseaux (p. ex. sans fil ou filaire) et la quantité optimale de dispositifs pour obtenir des résultats significatifs dans la prédiction d'un diagnostic cognitif.

Conclusion

Dans le cadre de nos travaux, nous avons proposé une méthodologie expérimentale rigoureuse d'acquisition et d'analyse de données pour le dépistage précoce de la démence de type Alzheimer dans un environnement intelligent. Rappelons qu'il n'existe actuellement aucune méthode de dépistage précoce de la DTA, qu'il semble exister des signes précurseurs au niveau cognitif, que ces signes seraient plus manifestes en situation de stress en milieu inconnu et que certaines tâches pourraient être discriminantes. C'est donc pour explorer ces questions que nous avons conçu cette méthodologie expérimentale. En premier lieu, nous avons mis en place un protocole d'expérimentation rigoureux pour assurer la qualité et la comparabilité des données acquises sur deux sites d'expérimentation. Ensuite, nous avons documenté et comparé les comportements des participants avec et sans troubles cognitifs lors de la réalisation d'AIVQ à l'aide du système d'acquisition de données hétérogènes que nous avons conçu, implémenté et déployé. Les données acquises ont été prétraitées en vue d'un apprentissage supervisé par classification automatique, où le but était de classer les participants selon leur diagnostic cognitif parmi les trois choix : cognitif sain, TCL ou DTA.

Nos analyses préliminaires indiquent que cela est possible. En effet, nos résultats permettent de déduire qu'il y a une différence significative entre la classe des personnes saines et celle des personnes TCL en utilisant un ensemble réduit de capteurs discriminants. Par contre, aucune différence significative n'a été observée entre la classe des personnes DTA et les autres classes. Nos analyses ont aussi permis d'observer une corrélation entre le temps requis pour compléter les tâches demandées et le diagnostic cognitif des participants. Finalement, ce projet vérifie la faisabilité de réaliser une expérimentation de cette envergure sur deux sites et la possibilité de générer des données comparables. Aussi, nous avons exploré la question du type et du nombre optimal de capteurs : les capteurs sans fil sont moins fiables que les filaires, mais peuvent tout de même être utilisés pour plusieurs types d'analyses; et un petit nombre de capteurs serait suffisant, pourvu qu'ils soient bien choisis.

Critique du travail

Ce projet de recherche représentait un travail considérable. À chaque étape du projet, nous nous sommes confrontés à une multitude de problèmes et dans la plupart des cas, plusieurs solutions étaient possibles, complexifiant ainsi le projet et exigeant un travail rigoureux et systématique. Rappelons que, à notre connaissance, aucune méthodologie comparable n'a été proposée auparavant. Ce travail visait donc entre autres à concevoir une méthodologie pour réaliser des expérimentations multisites et celle-ci a été testée sur deux sites d'expérimentations. Même si les deux sites étaient assez différents (disposition des pièces, caméras et capteurs utilisés, méthode d'annotation des données utilisée, etc.), les analyses montrent des résultats comparables, ce qui est prometteur.

Pour ce qui est du prétraitement et de l'analyse des données, la technique utilisée pour représenter les données de capteurs (les transformer sous forme de nombre de déclenchements de capteurs) est limitée. Cette représentation ne tient compte que de l'activité du participant et le temps requis pour compléter l'expérimentation. Elle est donc insensible au séquençage des actions et, selon nos résultats, ne nous permet pas de différencier les participants DTA des TCL. Malgré cela, cette analyse préliminaire a su confirmer que cette approche est tout de même prometteuse en obtenant une performance de classification de 67% avec l'algorithme OneR sur toutes les données après prétraitement et de 80% avec l'algorithme NaiveBayes en n'utilisant que les données des participants sains et TCL avec un sous-ensemble composé des cinq capteurs les plus discriminants. Ces résultats sont nettement supérieurs aux probabilités a priori (42% et 58% respectivement), ce qui est encourageant.

Travaux futurs de recherche

Cette maîtrise a exploré les bases nécessaires pour pousser la recherche future sur trois axes :

- Poursuivre l'analyse des données de capteurs du DOMUS en utilisant d'autres techniques et algorithmes d'apprentissage supervisé;

- Élaborer les analyses cliniques sur les données conjointes des laboratoires DOMUS et du CRIUGM;
- Aller plus loin dans l'étude de la faisabilité d'expérimentations multisites et de la quantité optimale de capteurs.

Une bonne partie des travaux futurs de recherche concerne l'étape d'analyse des données acquises lors des expérimentations. Nous avons effleuré l'idée, à la section 6.2.6, d'utiliser une technique de classification avec fenêtres temporelles qui permettrait d'avoir un plus grand nombre d'instances de classes pour la phase d'apprentissage. Cette technique fait partie de la voie poursuivie par Belkacem Chikhaoui du CRIM qui poussera l'analyse plus loin. De plus, il utilisera les annotations provenant de la tablette Android (décrites au Chapitre 4) pour découper les données dans le but d'isoler chaque tâche. Cela servira à analyser que les données concernant la tâche de préparation de repas qui est la plus discriminante. Bien entendu, d'autres algorithmes pourront être testés, comme les machines à vecteurs de support ou les réseaux de neurones. De plus, une attention particulière pourrait être portée au paramétrage de ces algorithmes.

La suite des analyses cliniques portera sur la recherche d'autres corrélations entre diverses variables observées et sur l'ajout de variables externes aux données d'apprentissage pour la classification. Ces variables externes seraient le temps de complétion, le nombre d'assistances, le nombre d'erreurs, etc. ; toutes des variables répertoriées dans CoLab lors de la saisie de verbatim.

Finalement, pour pousser plus loin l'étude de la faisabilité d'expérimentations multisites et de la quantité optimale de capteurs (les objectifs O2 et O3), plus d'expérimentations devront se dérouler au DOMUS ou sur d'autres sites pour pouvoir généraliser nos résultats et explorer plus en profondeur ces objectifs. Notre méthodologie fournira un cadre solide pour ces expérimentations. Elles permettront aussi d'obtenir plus de données utilisables pour de futures analyses en classification automatique.

Perspective

Nos travaux proposent et valident une méthodologie complète pour dépister la DTA de façon précoce sur différents sites d'expérimentation. Les données semblent transposables d'un lieu à l'autre. À terme, notre système d'acquisition de données et nos algorithmes pourront être déployés et utilisés à l'intérieur de cliniques spécialisées en dépistage de la DTA, par exemple dans certains CLSC ou hôpitaux du Québec. Les personnes qui auraient des doutes quant à leur santé cognitive (ou toute personne voulant être testée) viendraient accomplir les AIVQ demandées lors d'une expérimentation qui se déroulerait dans l'une ou l'autre de ces cliniques. Les données de capteurs accumulées lors de ces expérimentations pourraient être partagées pour continuer à alimenter et améliorer au fil du temps les algorithmes de classification.

Bibliographie

- [1] Agence de la santé publique du Canada, « Vol 36, No 10, octobre 2016 - Promotion de la santé et prévention des maladies chroniques au Canada : Recherche, politiques et pratiques - Agence de la santé publique du Canada ».
- [2] G. M. McKhann, « Changing Concepts of Alzheimer Disease », *JAMA*, vol. 305, n° 23, p. 2458, juin 2011.
- [3] R. A. Sperling, P. S. Aisen, L. A. Beckett, D. A. Bennett, S. Craft, A. M. Fagan, T. Iwatsubo, C. R. J. Jack, J. Kaye, T. J. Montine, D. C. Park, E. M. Reiman, C. C. Rowe, E. Siemers, Y. Stern, K. Yaffe, M. C. Carrillo, B. Thies, M. Morrison-Bogorad, M. V. Wagster, et C. H. Phelps, « Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging- Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease », *Alzheimer's Dement.*, vol. 7, n° 3, p. 280-292, 2011.
- [4] H. Amieva, H. Jacqmin-Gadda, J.-M. Orgogozo, N. Le Carret, C. Helmer, L. Letenneur, P. Barberger-Gateau, C. Fabrigoule, et J.-F. Dartigues, « The 9 year cognitive decline before dementia of the Alzheimer type: a prospective population-based study », *Brain*, vol. 128, n° 5, p. 1093-1101, mars 2005.
- [5] K. Peres, C. Helmer, H. Amieva, J. M. Orgogozo, I. Rouch, J. F. Dartigues, P. Barberger-Gateau, K. Pérès, C. Helmer, H. Amieva, J. M. Orgogozo, I. Rouch, J. F. Dartigues, et P. Barberger-Gateau, « Natural history of decline in instrumental activities of daily living performance over the 10 years preceding the clinical diagnosis of dementia: a prospective population-based study. », *J. Am. Geriatr. Soc.*, vol. 56, n° 1, p. 37-44, 2008.
- [6] É. Dutil, C. Bottari, M. Vanier, et J. Rousseau, *Profil des AVQ: version 4*. École des sciences de la réadaptation, Sciences de la santé, Université d'Ottawa, 2004.

- [7] D. J. Cook et S. K. Das, « How smart are our environments? An updated look at the state of the art », *Pervasive Mob. Comput.*, vol. 3, n° 2, p. 53-73, 2007.
- [8] J. Kaye, « Home-based technologies: A new paradigm for conducting dementia prevention trials », *Alzheimer's Dement.*, vol. 4, n° 1 SUPPL. 1, p. 60-66, 2008.
- [9] « Laboratoire Domus - Université de Sherbrooke ». [En ligne]. Disponible à: <https://www.usherbrooke.ca/domus/fr/>. [Consulté le: 16-juin-2017].
- [10] « Centre de recherche Institut universitaire de gériatrie de Montréal ». [En ligne]. Disponible à: <http://www.criugm.qc.ca/fr.html>. [Consulté le: 16-juin-2017].
- [11] « Z-Wave the Smartest Choice for your Smart Home ». [En ligne]. Disponible à: <http://www.z-wave.com/>. [Consulté le: 16-juin-2017].
- [12] « En quoi consistent l'Alzheimer et maladies apparentées | Société Alzheimer du Canada ». [En ligne]. Disponible à: <http://www.alzheimer.ca/fr/About-dementia/What-is-dementia>. [Consulté le: 09-sept-2017].
- [13] G. McKhann, D. Drachman, M. Folstein, R. Katzman, D. Price, et E. M. Stadlan, « Clinical diagnosis of Alzheimer's disease: Report of the NINCDS-ADRDA Work Group* under the auspices of Department of Health and Human Services Task Force on Alzheimer's Disease », *Neurology*, vol. 34, n° 7, p. 939-939, 1984.
- [14] « La maladie d'Alzheimer | Société Alzheimer du Canada ». [En ligne]. Disponible à: <http://www.alzheimer.ca/fr/About-dementia/Alzheimer-s-disease>. [Consulté le: 10-sept-2017].
- [15] H. Amieva, L. H. Phillips, S. Della Sala, et J. D. Henry, « Inhibitory functioning in Alzheimer's disease », *Brain*, vol. 127, n° 5, p. 949-964, 2004.

- [16] L. Bherer, S. Belleville, et C. Hudon, « [Executive function deficits in normal aging, Alzheimer's disease, and frontotemporal dementia] », *Psychol.Neuropsychiatr.Vieil.*, vol. 2, n° 1760-1703, p. 181-189, 2004.
- [17] F. Collette, M. Van Der Linden, et E. Salmon, « Executive dysfunction in Alzheimer's disease », *Cortex*, vol. 35, n° 1, p. 57-72, 1999.
- [18] R. J. Perry et J. R. Hodges, « Attention and executive deficits in Alzheimer's disease. A critical review », *Brain*, vol. 122, n° 3. p. 383-404, 1999.
- [19] M. M. Swanberg, R. E. Tractenberg, R. Mohs, L. J. Thal, et J. L. Cummings, « Executive dysfunction in Alzheimer disease. », *Arch. Neurol.*, vol. 61, n° 4, p. 556-60, 2004.
- [20] S. L. Willis, R. Allen-Burge, M. M. Dolan, R. M. Bertrand, J. Yesavage, et J. L. Taylor, « Everyday problem solving among individuals with Alzheimer's disease. », *Gerontologist*, vol. 38, n° 5, p. 569-577, 1998.
- [21] A. Miyake, N. P. Friedman, M. J. Emerson, A. H. Witzki, A. Howerter, et T. D. Wager, « The Unity and Diversity of Executive Functions and Their Contributions to Complex "Frontal Lobe" Tasks: A Latent Variable Analysis », *Cogn. Psychol.*, vol. 41, n° 1, p. 49-100, 2000.
- [22] H. Amieva, H. Jacqmin-Gadda, J. M. Orgogozo, N. Le Carret, C. Helmer, L. Letenneur, P. Barberger-Gateau, C. Fabrigoule, et J. F. Dartigues, « The 9 year cognitive decline before dementia of the Alzheimer type: A prospective population-based study », *Brain*, vol. 128, p. 1093-1101, 2005.
- [23] P. Dubreuil, S. Adam, N. Bier, et L. Gagnon, « The ecological validity of traditional memory evaluation in relation with controlled memory processes and routinization », *Arch. Clin. Neuropsychol.*, vol. 22, n° 8, p. 979-989, 2007.

- [24] J. Kaye, N. Mattek, H. Dodge, T. Buracchio, D. Austin, S. Hagler, M. Pavel, et T. Hayes, « One walk a year to 1000 within a year: continuous in-home unobtrusive gait assessment of older adults. », *Gait Posture*, vol. 35, n° 2, p. 197-202, 2012.
- [25] J. Bouisson, « Routinization preferences, anxiety, and depression in an elderly French sample », *J. Aging Stud.*, vol. 16, n° 3, p. 295-302, 2002.
- [26] J. Bouisson et J. Swendsen, « Routinization and Emotional Well-Being: An Experience Sampling Investigation in an Elderly French Sample. », *Journals Gerontol. Ser. B Psychol. Sci. Soc. Sci.*, vol. 58B, n° 5, p. P280-P282, 2003.
- [27] A. Akl, B. Taati, et A. Mihailidis, « Autonomous unobtrusive detection of mild cognitive impairment in older adults », *IEEE Trans. Biomed. Eng.*, vol. 62, n° 5, p. 1383-1394, 2015.
- [28] H. H. Dodge, N. C. Mattek, D. Austin, T. L. Hayes, et J. A. Kaye, « In-home walking speeds and variability trajectories associated with mild cognitive impairment », *Neurology*, vol. 78, n° 24, p. 1946-1952, 2012.
- [29] P. N. Dawadi, D. J. Cook, M. Schmitter-Edgecombe, et C. Parsey, « Automated assessment of cognitive health using smart home technologies. », *Technol. Health Care*, vol. 21, n° 4, p. 323-43, 2013.
- [30] C. L. Bottari, C. Dassa, C. M. Rainville, et E. Dutil, « The IADL Profile: development, content validity, intra- and interrater agreement [corrected] [published erratum appears in CAN J OCCUP THER 2010 Jun;77(3):134] », *Can. J. Occup. Ther.*, vol. 77, n° 2, p. 90-100, 2010.
- [31] « Interface de programmation ». [En ligne]. Disponible à: https://fr.wikipedia.org/wiki/Interface_de_programmation. [Consulté le: 27-juill-2017].

- [32] « Un appartement intelligent - Laboratoire Domus - Université de Sherbrooke ». [En ligne]. Disponible à: <https://www.usherbrooke.ca/domus/fr/recherche/infrastructure/un-appartement-intelligent/>. [Consulté le: 27-juill-2017].
- [33] « Official download of VLC media player, the best Open Source player - VideoLAN ». [En ligne]. Disponible à: <https://www.videolan.org/vlc/index.html>. [Consulté le: 06-sept-2017].
- [34] « SQLite Home Page ». [En ligne]. Disponible à: <https://www.sqlite.org/>. [Consulté le: 13-juin-2017].
- [35] « Microsoft Data Platform | Microsoft ». [En ligne]. Disponible à: <https://www.microsoft.com/en-us/sql-server/default.aspx>. [Consulté le: 06-sept-2017].
- [36] « APAX-6572-AE-PC-based Controller w/ Atom ». [En ligne]. Disponible à: <https://buy.advantech.com/programmable-controllers-pacs/apax-controllers-pc-based-controller/model-apax-6572-ae.htm>. [Consulté le: 11-sept-2017].
- [37] « Vera Smarter Home Control ». [En ligne]. Disponible à: <http://getvera.com/>. [Consulté le: 06-sept-2017].
- [38] « OpenRemote | Open Source for Internet of Things ». [En ligne]. Disponible à: <http://www.openremote.com/>. [Consulté le: 06-sept-2017].
- [39] « Build Apps with JavaScript | Meteor ». [En ligne]. Disponible à: <https://www.meteor.com/>. [Consulté le: 10-sept-2017].
- [40] « DB Browser for SQLite ». [En ligne]. Disponible à: <http://sqlitebrowser.org/>. [Consulté le: 11-sept-2017].
- [41] « PowerShell Documentation | Microsoft Docs ». [En ligne]. Disponible à: <https://docs.microsoft.com/fr-fr/powershell/>. [Consulté le: 11-sept-2017].

- [42] « Welcome to Python.org ». [En ligne]. Disponible à: <https://www.python.org/>. [Consulté le: 11-sept-2017].
- [43] I. H. Witten, E. Frank, et M. A. Hall, « Weka 3 - Data Mining with Open Source Machine Learning Software in Java », 2011. [En ligne]. Disponible à: <http://www.cs.waikato.ac.nz/ml/weka/>. [Consulté le: 19-juill-2017].
- [44] « Real Time Streaming Protocol ». [En ligne]. Disponible à: https://en.wikipedia.org/wiki/Real_Time_Streaming_Protocol. [Consulté le: 11-sept-2017].
- [45] « HTML5 Video ». [En ligne]. Disponible à: https://www.w3schools.com/html/html5_video.asp. [Consulté le: 11-sept-2017].
- [46] « Windows Movie Maker ». [En ligne]. Disponible à: https://en.wikipedia.org/wiki/Windows_Movie_Maker. [Consulté le: 11-sept-2017].
- [47] « Avidemux - Main Page ». [En ligne]. Disponible à: <http://avidemux.sourceforge.net/>. [Consulté le: 11-sept-2017].
- [48] S. L. Salzberg, « C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993 », *Mach. Learn.*, vol. 16, n° 3, p. 235-240, sept. 1994.
- [49] « J48 ». [En ligne]. Disponible à: <http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html>. [Consulté le: 19-juill-2017].
- [50] « Cross-validation (statistics) ». [En ligne]. Disponible à: [https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics)). [Consulté le: 25-août-2017].
- [51] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. Mclachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, D. Steinberg, X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, et H. Motoda, « Top 10 algorithms in data mining », *Knowl Inf Syst*, vol. 14, p. 1-37, 2008.

- [52] L. Breiman, « Random Forests », *Mach. Learn.*, vol. 45, n° 1, p. 5-32, 2001.
- [53] F. Bergeron, K. Bouchard, S. Gaboury, S. Giroux, et B. Bouchard, « Indoor Positioning System for Smart Homes Based on Decision Trees and Passive RFID », dans *20th Pacific Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD) 2016*, 2016, p. 42-53.
- [54] G. H. G. John et P. Langley, « Estimating Continuous Distributions in Bayesian Classifiers », *Proc. Elev. Conf. Uncertain. Artif. Intell. Montr. Quebec, Canada*, vol. 1, p. 338--345, 1995.
- [55] « Théorème de Bayes ». [En ligne]. Disponible à : https://fr.wikipedia.org/wiki/Théorème_de_Bayes. [Consulté le: 29-août-2017].
- [56] R. C. Holte, « Very Simple Classification Rules Perform Well on Most Commonly Used Datasets », *Mach. Learn.*, vol. 11, n° 1, p. 63-90, 1993.
- [57] J. C. Platt, « Fast training of support vector machines using sequential minimal optimization », *Adv. kernel methods*, p. 376, 1999.
- [58] V. N. Vapnik, *The nature of statistical learning theory*. 1995.
- [59] « Sélection des fonctionnalités (exploration de données) ». [En ligne]. Disponible à : [https://msdn.microsoft.com/fr-fr/library/ms175382\(v=sql.120\).aspx](https://msdn.microsoft.com/fr-fr/library/ms175382(v=sql.120).aspx). [Consulté le: 01-sept-2017].
- [60] M. Lussier, S. Giroux, B. Chikhaoui, M. Gagnon, S. Adam, C. Consel, J. Macoir, I. Rouleau, L. Talbot, et N. Bier, « Memory, executive function and naturalistic assessment of activities of daily living in mild cognitive impairment using smart home sensors (en préparation) », 2017.

Annexe A

Capteurs sans fil

A.1 Liste des capteurs

Cette section présente la liste des capteurs sans fil utilisés au cours de cette maîtrise.

A.1.1 Convention pour les noms des capteurs sans fil

- Premier caractère
 - Z pour Z-Wave
- Deuxième caractère
 - C pour Capteur de contact
 - M pour Capteur de mouvement
 - ESS pour Capteur de consommation électrique
- Troisième et quatrième caractères
 - CH pour Chambre
 - CU pour Cuisine
 - EN pour Entrée
 - SB pour Salle de bain
 - SL pour Salon
 - SM pour Salle à manger

A.1.2 Capteurs de contact électromagnétique

Capteurs	Capteurs utilisés	Détails et défaillances connues
Chambre		
ZCCH	01 : portes d'armoire à droite du lit	Un seul capteur utilisé pour les deux portes, car on ne va rarement en n'ouvrir qu'une seule
	02 : premier tiroir du haut à droite du lit	Tiroir dans lequel il faut placer les vêtements lors des expérimentations du projet dépistage
Cuisine		
ZCCU	01-04 : portes d'armoire au-dessus de la cuisinière, en ordre de gauche à droite	01 : contient le sel/sucre (nous n'avions pas de poivre), le café et les filtres 02-03 : ne contient rien d'important 04 : contient les assiettes, bols, etc.
	05-07 : portes d'armoire à droite de la cuisinière	05 : porte en coin, ne contient rien d'important 06-07 : 06 (tasses) et 07 (verres) sont faits en verre et recouvert d'un VarioGlass, comme la « fenêtre » au-dessus de l'évier à proximité. En plus, les poignées sont difficiles à localiser et à utiliser
	08 : porte d'armoire en haut à droite de l'évier, à côté du frigo	Rien d'important dans cette armoire
	09; 10 : portes d'armoire au-dessus du four à micro-ondes/ cafetière/ grille-pain	Un seul capteur par portes doubles, car on ne va rarement en n'ouvrir qu'une seule. Ne contient rien d'important 09 : contient divers thés et autres boissons (pas de café), inutiles lors des expérimentations du projet dépistage

		10 : contient des objets divers inutiles lors des expérimentations du projet dépistage
	11; 12/18/19; 13; 14 : tiroirs à ustensiles, situés entre l'évier et le frigo	11 : premier tiroir, contient les couverts (fourchettes, cuillères, couteaux) 12/18/19 : deuxième tiroir, contient les autres ustensiles (spatule, grands couteaux, louche, ciseaux, etc.). A changé de numéro au fil du temps, car nous avons eu des problèmes avec le capteur 13 : troisième tiroir, ne contient rien d'important 14 : quatrième tiroir, ne contient rien d'important
	15 : portes d'armoire en-dessous de l'évier	Un seul capteur par portes doubles, car on ne va rarement en n'ouvrir qu'une seule. Contient la poubelle à compost et la poubelle normale (toutes deux identifiées comme telles) et divers produits nettoyants (l'éponge à récurer et le savon à vaisselle sont déjà à côté de l'évier)
	16 : porte d'armoire sous le grille-pain	Contient les casseroles et poêlons. Malheureusement, ce capteur a fait défaut la plupart du temps lors des expérimentations du projet dépistage
Entrée		
ZCEN	01 : capteur de la porte d'entrée	C'est le premier capteur à se déclencher pour TOUTES les expérimentations du projet dépistage. Malheureusement, ce capteur a fait défaut plusieurs fois.
Salle de bain		
ZCSB	01 : porte d'armoire en bas à gauche de l'évier	C'est la seule porte / tiroir qu'il est nécessaire d'ouvrir lors des expérimentations du projet dépistage

A.1.3 Capteurs de mouvement

Capteurs	Capteurs utilisés
Chambre	
ZMCH	01 : dans un coin en haut du mur, près de la garde-robe. Placé pour remarquer la personne entrer et la voir peu importe ce qu'elle fait dans la chambre
Cuisine	
ZMCU	01 : dans un coin en haut du mur, près des armoires à proximité du four à micro-ondes. Placé pour remarquer toute activité dans la cuisine
Entrée	
ZMEN	01 : dans un coin en haut du mur, près de la porte d'entrée. Placé pour remarquer la personne lorsqu'elle entre et lorsqu'elle place ses vêtements dans la garde-robe
Salle de bain	
ZMSB	01 : dans un coin en haut du mur, près de la porte d'entrée de la salle de bain. Placé pour remarquer la personne entrer et effectuer la tâche demandée
Salon	
ZMSL	01 : au plafond. Placé pour réagir aux mouvements de la personne dans le salon sans « voir » les expérimentateurs
Salle à manger	
ZMSM	01 : au plafond, au-dessus de la table. Placé pour voir la table à manger et autour, dont en bordure de la cuisine et du salon (ce qui était nécessaire lors des expérimentations du projet dépistage)

A.1.4 Capteurs de consommation électrique

Capteurs	Capteurs utilisés
Cuisine	
ZESSCU	01 : capteur de la cafetière 02 : capteur du grille-pain

A.2 Sous-ensemble de capteurs nécessaires pour les expérimentations du projet DEI

Cette section présente la liste des capteurs sans fil qui sont jugés comme étant nécessaires pour la complétion de chaque tâche demandée lors d'une expérimentation du projet DEI.

A.2.1 Capteurs de contact électromagnétique

Capteurs	Nombre de déclenchements nécessaires	Explication
Chambre		
ZCCH 02	1 on 1 off	Ouvrir le tiroir pour placer les vêtements pliés et le refermer.
Cuisine		
ZCCU 01	2 on 2 off	Ouvrir et fermer pour sortir le sucre, le café et les filtres, puis pour les ranger. Ouvrir aussi pour ajouter du sel aux œufs si désiré (optionnel). Sera ouvert probablement plus de deux fois.

ZCCU 04	2 on 2 off	Ouvrir et fermer pour sortir une assiette et pour la ranger.
ZCCU 06	2 on 2 off	Ouvrir et fermer au minimum 2 fois pour prendre et ranger une tasse. L'armoire surveillé par le capteur ZCCU 07 pouvait à l'occasion contenir des tasses, donc le participant peut en avoir pris une là et jamais avoir ouvert la porte ZCCU 06.
ZCCU 11	2 on 2 off	Ouvrir et fermer au minimum 2 fois pour sortir les couverts (cuillère, couteau à beurre, fourchette) et pour les ranger.
ZCCU 12/18/19	2 on 2 off	Ouvrir et fermer au minimum 2 fois pour sortir une spatule ou autre ustensile utile pour faire cuire les œufs, puis le ranger. Peut être optionnel si la personne utilise une casserole pour la cuisson des œufs.
ZCCU 15	1 on 1 off	Ouvrir au minimum une fois pour jeter le restant de repas dans le compost. Sera ouvert probablement plus d'une fois.
ZCCU 16	2 on 2 off	Ouvrir pour prendre une casserole ou un poêlon pour faire cuire les œufs, puis refermer. Ouvrir plus tard pour ranger.
Entrée		
ZCEN 01	1 on 1 off	Ouvrir la porte pour entrer dans l'appartement, puis la refermer en sortant. L'ouverture de la porte marque le début de l'expérimentation, mais pas la fin.
Salle de bain		
ZCSB 01	1 on 1 off	Ouvrir la porte d'armoire où se trouve tous les produits et outils nécessaire à nettoyer la salle de bain (produits nettoyants, tampons à récurer, gants, etc.).

A.2.2 Capteurs de mouvement

Tous les six capteurs de mouvement sont nécessaires à la réussite de l'expérimentation.

A.2.3 Capteurs de consommation électrique

Tous les deux capteurs de consommation électrique sont nécessaires à la réussite de l'expérimentation.

Annexe B

Listes de vérification lors des expérimentations

B.1 Liste préexpérimentation

Liste de conditions initiales à vérifier avant le début de chaque expérimentation. La colonne titrée « EXP23 » sert d'exemple. Pour une expérimentation subséquente, nous aurions une nouvelle colonne à la droite de celle-ci nommée par exemple « EXP24 ». Le terme « OK » sur fond vert signifie que la condition est vérifiée. « NON » sur fond rouge signifie qu'elle ne l'est pas. La couleur de fond change automatiquement selon le texte entré.

Tâche	EXP23
CONDITIONS INITIALES	NON
CUISINE	NON
Réorganiser le frigo pour l'expérimentation: s'assurer qu'il y ait : 1 pot de margarine, 1 pot de confiture, 1 contenant de lait, du pain, des œufs.	OK
Mettre en place le grille-pain et la (petite) cafetière en les branchant dans les capteurs de courant Z-Wave (Energy Switch) sur le comptoir où se situe le micro-ondes, côte à côte. S'assurer qu'ils sont propres et bien configurés (vérifier puissance du grille-pain).	NON
Dans l'armoire en bas à droite du comptoir où se situe le four à micro-ondes (capteurs CC6-CC7), placer les poêlons et les casseroles	OK
Dans l'armoire du haut à gauche de la cuisinière (capteur CA1), on doit retrouver : le café, les filtres et le sucre.	OK
Dans l'armoire du haut à droite de la cuisinière (capteur CA4), on retrouve les assiettes, bols, etc.	OK
Dans l'armoire vitrée de gauche (capteur CA6), on y retrouve les tasses.	OK
Dans le tiroir du haut juste à gauche du frigo (capteur CB12), on y trouve les ustensils (couverts)	OK
Dans le 2e tiroir du haut juste à gauche du frigo (capteur CB13), on y trouve les ustensils du genre: spatule, fouet, grands couteaux, etc.	OK

Placer la petite poubelle sous l'évier	OK
Laver et ranger la vaisselle	OK
Vider l'égouttoir à vaisselle	OK
Replacer les napperons (visibles sur le comptoir où est le micro-ondes, près du grille-pain)	OK
Vérifier le bon fonctionnement de la cuisinière	OK
Placer un panneau "Lave-Vaisselle HORS-SERVICE"	OK
Enlever la bouilloire	OK
Enlever tout post-it ou feuille placée sur les armoires	OK
Enlever tous les couverts et ustensiles en plastique	OK
Enlever tout savon, shampoing ou produit nettoyant superflu	OK
S'assurer que l'éponge à récurer présente à côté de l'évier soit propre (l'échanger pour une autre si nécessaire)	OK
Laver le comptoir, le micro-ondes et la table de cuisine	OK
S'assurer que la caméra SONY du plafond regarde dans la bonne direction (mettre une config par défaut)	OK
Placer la caméra surélevée sur le contenant de plastique de sorte qu'on puisse voir le visage du sujet à la cuisinière et dans la majeure partie de la cuisine	OK
Désactiver le détecteur de fumée	OK
SALLE DE BAIN	OK
Dans la salle de bain, dans l'armoire à gauche de l'évier, on doit y retrouver : produits nettoyants, chiffon, essuie-tout, gants de nettoyage (optionnel)	OK
Enlever tout savon, shampoing ou produit nettoyant superflu	OK
Placer la caméra sur le trépied (complètement déplié, au-dessus du bain, les pieds du trépied reposant sur les côtés du bain) pour couvrir la zone du lavabo jusqu'à presque voir la toilette	OK
SALON	OK
Enlever le tapis	OK
Tirer les rideaux (descendre les stores)	OK
Vérifier qu'il n'y a pas déjà des éléments superflus apportés par Christine dans l'appartement	OK
Étiquette sur le téléphone "Pour composer un numéro, faites le 9 + numéro"	OK

Installer porte crayon et papier pour prise de notes près du téléphone, ainsi que l'annuaire tout près (bottin).	OK
Vérifier que le volume du téléphone est assez fort	OK
Enlever (ou cacher) les périphériques d'ordinateur en-dessous du Smartboard	OK
Placer et allumer la lampe de chevet près du bottin	OK
Placer la caméra sur le trépied (complètement déplié) au fond de la salle à manger, vis-à-vis les X gris	OK
CHAMBRE	OK
Dans la chambre, enlever/cacher les boîtes et les trucs qui traînent	OK
Placer les vêtements sur le lit de la chambre et 1-2 morceau dans le 1er tiroir de l'armoire de droite	OK
Préparer la caméra de la chambre (vérifier angle de vision)	OK
ENTRÉE	OK
Cacher les fils du contrôleur Vera, s'il est présent	OK
Ajouter quelques manteaux dans la garde-robe d'entrée pour qu'il ressemble plus à un garde-robe	OK
Ajouter une affiche "Expérimentation en cours" sur la porte d'entrée du DOMUS	OK
Ajouter une affiche "Expérimentation en cours" sur la porte d'entrée de la salle des serveurs, au cas où les techniciens viendraient	OK
ACQUISITION / CONSIDÉRATIONS TECHNIQUES	NON
AVANT L'EXP	NON
Prendre en note s'il y a des capteurs, caméras, micros, etc. qui sont dysfonctionnels pour cette EXP	NON
Éteindre les lumières "chambre, salle de bain et salle à manger (RL25, RL26, RL22)" si la porte de l'appart est ouverte pour la mise en place des fonctionnalités Reflex (étape suivante)	OK
Mettre en place les fonctionnalités Reflex de WinAppApax pour jouer un son "beep" et allumer les lumières : "chambre, salle de bain, salle à manger (RL25, RL26, RL22), RL34, RL43" à l'ouverture de la porte du DOMUS (voir fichier ApaxReflex.xml sur le bureau de l'APAX)	OK
Tester toutes les caméras IP et vérifier l'angle de vision	OK
JUSTE AVANT DE COMMENCER	OK
Démarrer enregistrement vidéo caméra Sony et autres caméras IP avec VLC --> avec le script de Jules	OK

Démarrer chacun des caméscopes (chambre [dlink], salle de bain [canon], cuisine [christine], salle à manger [sony])	OK
Allumer les 4 contrôleurs audio dans la salle du serveur et démarrer l'acquisition du son sur MatLab (exécuter script main.m)	OK
Lancer la récupération des données du DOMUS (créer un scénario avec date et # sujet, puis démarrer scénario). Pour ce faire, ouvrir WindowViewer sur Dinf-Duc. Minimiser avec Windows+D lorsque démarré.	OK
Lancer la récupération des données Z-Wave (sur le contrôleur Vera) → démarrer OpenRemote et Postgresql sur Dinf-duc	OK
Lancer l'application sur la tablette	OK
LA VEILLE	OK
Mettre la tablette sur la charge	OK
Mettre le caméscope sur la charge	OK
Vérifier l'état des capteurs avec l'interface du contrôleur Vera et OpenRemote	OK
Test de son	OK

B.2 Liste post-expérimentation

Liste de vérifications à faire après chaque expérimentation. La colonne titrée « EXP23 » sert d'exemple. Pour une expérimentation subséquente, nous aurions une nouvelle colonne à la droite de celle-ci nommée par exemple « EXP24 ». Le terme « OK » sur fond vert signifie que la condition est vérifiée. « NON » sur fond rouge signifie qu'elle ne l'est pas. La couleur de fond change automatiquement selon le texte entré.

Tâche	EXP23
DANS L'APPARTEMENT	NON
CUISINE	OK
Arrêter la caméra de la cuisine et vérifier l'état de la pile	OK
Copier les fichiers video	OK
Réactiver le détecteur de fumée	NON

SALLE DE BAIN	OK
Arrêter la caméra de la salle de bain et vérifier l'état de la pile	OK
Copier les fichiers vidéo du dossier \PRIVATE\AVCHD\BDMV\STREAM (caméra Canon) à l'endroit où l'on stocke les vidéos d'expérimentations	OK
Mettre la caméra sur la charge	OK
Enlever les produits de nettoyage, chiffon, essuie-tout, etc. et les ranger dans la boîte	OK
SALON / SALLE À MANGER	OK
Arrêter la caméra de salle à manger et vérifier l'état de la pile	OK
Copier les fichiers vidéo à l'endroit où l'on stocke les vidéos d'expérimentations	OK
Mettre la caméra sur la charge	OK
Ranger l'annuaire, papier, crayon dans la boîte (seulement après la dernière EXP de la journée)	OK
CHAMBRE	OK
Arrêter la caméra de la chambre (copie des fichiers automatique)	OK
Sortir les vêtements du tiroir et les ranger dans la boîte (seulement après la dernière EXP de la journée)	OK
ENTRÉE	OK
Enlever les affiches "Défense d'entrer" de la porte d'entrée et de la salle du serveur	OK
ACQUISITION / CONSIDÉRATIONS TECHNIQUES	OK
GÉNÉRAL	OK
Terminer le scénario sur la tablette (si ce n'est déjà fait) puis arrêter l'application	OK
Mettre la tablette sur la charge (la brancher dans l'ordi)	OK
Rouler le script sqlite_get.bat pour copier la BD SQLite localement et renommez le fichier selon le standard	OK
Arrêter enregistrement vidéo caméra IP avec VLC	OK
Arrêter la récupération des données du DOMUS	OK
Arrêter la récupération des données Z-Wave (sur le contrôleur Vera)	OK
Désactiver fonctionnalités Reflex (seulement après la dernière EXP de la journée)	OK
Insérer nom du scénario dans DB postgres	OK

SON	OK
Arrêter l'enregistrement du son sur le portable (arrêter script main.m)	OK
Démarrer la conversion du son en spécifiant le nom du dossier contenant les fichiers .txt au script de conversion (PlotDataSound.m)	OK
Copier les fichiers sonores convertis à l'endroit où l'on stocke les fichiers audio d'expérimentations	OK
Éteindre les 4 contrôleurs audio dans la salle du serveur	OK
VÉRIFICATION DES DONNÉES	OK
Prendre en note s'il y a des capteurs, caméras, micros, etc. qui ont été dysfonctionnels pour cette EXP (ceux qu'on a remarqué pendant ou après l'expé)	OK
Spot-check fichiers sonores (écouter quelques extraits pour vérifier que tout a fonctionné)	OK
Spot-check vidéos caméscopes et IP (regarder quelques extraits pour vérifier que tout a fonctionné)	OK
Vérifier log OpenRemote et insertion dans la BD	OK
Vérifier BD SQLite de la tablette Android	OK
Vérifier BD Enregistrement_capteurs (MS SQL Server du DOMUS)	OK

Annexe C

Scripts utilisés dans le projet

Cette section présente les principaux scripts utilisés pour diverses tâches au cours du projet.

C.1 Copier la base de données SQLite interne d'Android

Le script `sqlite_get.bat` sert à copier la BD SQLite de l'application APD pour pouvoir utiliser les données.

```
1 rem Change the following path to your own Android SDK path
2 D:
3 cd D:\Programmes\DOMUS-adt-bundle-windows-x86_64-20140702\sdk\platform-tools
4
5 rem Run this shell command to copy the db from the inner memory of your Android
  device (/data/data/...) to a public location (/sdcard/...). Modify these paths and
  your application package name accordingly.
6 adb -d shell "run-as ca.usherbrooke.domus.annotationprojetdepistage cat
  /data/data/ca.usherbrooke.domus.annotationprojetdepistage/databases/AnnotationProjet
  Depistage > /sdcard/Documents/annotation/AnnotationProjetDepistage.db
7
8 rem Set the file name relatively to the time the script is run
9 rem cut off fractional seconds
10 set t=%time:~0,5%
11 rem replace colons with dashes
12 set t=%t:=-%
13 set FileName=%date%_%t%_EXP
14
15 rem This command will copy the file from the public space of your Android device
  to a specified folder on your Windows computer. Modify these paths accordingly.
16 adb pull /sdcard/Documents/annotation/AnnotationProjetDepistage.db C:\annotation\
  %FileName%.db"
17 C:
18 pause
```

Figure 28 – Script qui copie la base de données SQLite interne d'Android

C.2 Concaténer les données en un seul fichier (concatenate_exp)

Utiliser le script `concatenate_exp` suivant, codé en Python avec le logiciel PyCharm, pour concaténer les données de toutes les expérimentations en un seul fichier.

Exécutez le script en vous assurant qu'aucun autre `.CSV` que ceux voulus n'est présent dans le dossier spécifié par la variable `inputfilepath`. Ensuite, remplacez la première ligne du fichier ("`#Type ...`") par la première ligne d'un des fichiers de données, la ligne décrivant les capteurs ("`ID", "...`"). Cette manœuvre est faite manuellement, car je n'ai pas réussi à la faire faire par le script avec le temps que j'avais.

```
import os

liste = []
inputfilepath = "[...]/DONNEES_FILAIRES/02. TRONQUES/"
outputfilepath = "[...]/DONNEES_FILAIRES/03. CONCATENEES/AllexPs.csv"

for f in os.listdir(inputfilepath):
    if f.endswith(".csv"):
        fichier = open(inputfilepath + f, 'rb')
        for line in fichier:
            if line.startswith('ID'):
                continue
            liste.append(line)
            liste.append("\n")

str = ''.join(liste)

destination = open(outputfilepath, 'wb')
destination.write(str)
destination.close()
```

Figure 29 – Script pour concaténer les données des expérimentations en un seul fichier

C.3 Transformer les décimaux à virgule en décimaux à point

Cette manœuvre est nécessaire pour rendre les données compatibles avec le logiciel Weka, car sinon il reconnaît les décimaux à virgule comme étant des *String* ou des *Nominal*. Pour ce faire, la méthode utilisée est la suivante:

Ouvrir le fichier de données dans un éditeur (comme Notepad++ ou Atom) et faire un **Search and Replace** (en mode **Regular Expression / Expression régulière**) :

- Écrire ceci dans la barre de recherche :
 - `(["']\d+),(\d+["'])`
 - Traduction : Guillemets doubles ou simples, suivi d'un caractère quelconque, suivi d'une virgule, suivi d'une suite de chiffres, puis de guillemets doubles ou simples.
- Écrire ceci dans la barre de remplacement :
 - `$1.$2`
 - Groupe RegEx 1, suivi de point, suivi du groupe RegEx 2
 - Les groupes sont entre parenthèses dans l'expression régulière
 - ATTENTION : la procédure est longue (~2mins) et peut faire geler l'application. Si ça arrive, réessayer. Si encore, essayez avec un autre éditeur de texte.

C.4 Binariser les valeurs des débitmètres

Utiliser le script **standardizeFlowmeter**, codé en R dans Rstudio, en spécifiant le bon fichier de données (en entrée et en sortie) et les expérimentations qu'on veut extraire de ce fichier. Toute valeur de débitmètre plus grande ou égale à 10 sera remplacée par 1, et les 0 resteront 0 (il n'y a pas de valeur entre 0 et 10). Nous aurons donc une donnée simplifiée du style "ouvert/fermé". Le script nous produira un fichier "outputDebitmetre.csv" contenant les données nouvellement formatées. Les valeurs des capteurs de contact et de mouvement sont inversées par ce script pour être plus logiques et pour avoir le même sens que les données sans-fil et celles de Montréal.

```
#Data Frame containing all data
inputFile = "[...]/DONNEES_FILAIRES/05. APRES_filtres/AlLEXPs-
27janv2017-apresFiltres.csv"
outputFile = "[...]/DONNEES_FILAIRES/06. DEBITMETRES
BINARISES/outputDebitmetre30janv2017.csv"

allData = read.csv(inputFile)
```

```

expList <- c('EXP18', 'EXP21', 'EXP22', 'EXP23', 'EXP24', 'EXP25',
'EXP26', 'EXP27', 'EXP28', 'EXP29', 'EXPDTA16', 'EXPDTA17',
          'EXPDTA18', 'EXPDTA19', 'EXPDTA20', 'EXPMCI16',
'EXPMCI17', 'EXPMCI18', 'EXPMCI19', 'EXPMCI20', 'EXPMCI21',
          'EXPMCI23', 'EXPMCI24', 'EXPMCI25', 'EXPMCI26')

###Ajouter une premiere ligne qui contient les titres des colonnes
write(colnames(expData),
      file = outputFile,
      ncolumns = length(colnames(expData)),
      sep = ",")

#Loop over all EXPs
for (i in 1:length(expList)){
  #Extract data for the current EXP
  expData = allData[grep(expList[i], allData$No_Test),]
  dbSensorList = grep("DB", colnames(expData[,2:length(expData[1,])]),
value = TRUE)
  mvSensorList = grep("MV", colnames(expData[,2:length(expData[1,])]),
value = TRUE)
  ccSensorList = grep("C", colnames(expData[,2:length(expData[1,])]),
value = TRUE)

  #for all lines of data in the EXP
  for (j in 1:length(expData[,1])){
    for (dbSensor in dbSensorList){
      if (expData[[dbSensor]][j] >= 10) {expData[[dbSensor]][j] <- 1}
    }

    #invert values of movement sensors
    for (mvSensor in mvSensorList){
      if (expData[[mvSensor]][j] == 1) {
        expData[[mvSensor]][j] <- 0
      } else{
        expData[[mvSensor]][j] <- 1
      }
    }
  }

  #invert values of contact sensors
  for (ccSensor in ccSensorList){
    if (expData[[ccSensor]][j] == 1) {
      expData[[ccSensor]][j] <- 0
    } else{
      expData[[ccSensor]][j] <- 1
    }
  }
}

write.table(expData, file = outputFile,
            row.names = FALSE, col.names = FALSE, append = TRUE,
            qmethod = "double", dec = ".", sep = ",")
}

```

Figure 30 – Script qui binarise les valeurs des débitmètres

Annexe D

Critères de sélection des participants

D.1 Participants sans troubles cognitifs

Les critères d'inclusion sont : 1) être âgé de 65 ans ou plus; 2) obtenir des résultats normaux aux tests de dépistage des troubles cognitifs selon l'âge et le niveau de scolarité, tels que mesurés par le Montreal Cognitive Assessment (MOCA) (Nasreddine et al., 2005). Les critères d'exclusion sont reliés aux conditions de santé pouvant produire des troubles cognitifs : 1) Historique d'atteinte cérébrale (traumatisme crânien, accident vasculaire cérébral, encéphalopathies); 2) Diabète ou hypertension non contrôlés; 3) Présence de troubles psychiatriques : schizophrénie, troubles bipolaires, anxiété ou dépression. Les personnes avec des symptômes dépressifs subcliniques, mesurés avec le Beck Depression Inventory (BDI) (Beck, 1997) ne seront pas exclues; 4) Délirium dans les six derniers mois; 5) Chirurgie intracrânienne; 6) Déficience en vitamine B12 ou éthyliisme; 7) Utilisation d'une médication qui peut influencer la cognition et la vigilance (hypnotiques, neuroleptiques, ou anticonvulsivants); 8) Incapacités physiques limitant la capacité à se déplacer seul et de façon sécuritaire dans l'appartement intelligent. Trois tests mesurant les fonctions exécutives seront administrés aux participants : Trail Making Test A et B (Reitan, 1958), Stroop et Test de la tour de la batterie D-KEFS (Delis, Kaplan, & Kramer, 2001). Seront également administrés, afin d'avoir un portrait global des participants et de tenir compte de plusieurs facteurs qui pourraient influencer de façon importante la performance aux activités quotidiennes, dont le temps de réalisation (une de nos principales variables mesurées par la technologie) : deux tests de mémoire épisodique (les 15 mots de Rey ainsi qu'un test de mémoire de la source - protocole maison), deux tests de langage (analyse du langage spontanée et connaissances des mots et des actions impliquées dans les tâches quotidiennes effectuées lors du scénario), un test de réserve cognitive (Cognitive Reserve Index questionnaire; Nucci, Mapelli & Mondini, 2012), un questionnaire de dépistage sur les risques de chutes (Falls Efficacy Scale-International – FES-

I; Yardley et al., 2005), un questionnaire sur la qualité du sommeil et une évaluation des troubles visuels. Certains de ces questionnaires pourront être remplis par les participants à leur domicile au besoin, pour alléger la rencontre (ex. chutes, réserve cognitive).

D.2 Participants avec TCL

Le diagnostic de TCL sera confirmé par les milieux cliniques collaborant à ce projet [cliniques externes de l'Institut universitaire de gériatrie de Montréal (IUGM) et du CSSS-Institut universitaire de gériatrie de Sherbrooke (IUGS)], en accord avec les critères actuels de TCL (Petersen, et al., 2001; Winblad et al., 2004). Les sujets présentant un TCL-amnésique et à domaines multiples seront recrutés. Les critères d'exclusion seront les mêmes que pour ceux des sujets sains.

D.3 Participants avec DTA

Les participants avec troubles cognitifs présenteront un diagnostic de DTA dans les stades débutants de la maladie [stades 2-4; Sclan et Reisberg (1992)]. Le diagnostic ne sera pas émis dans le cadre de cette étude, mais sera confirmé par les équipes multidisciplinaires qui participeront au recrutement des participants. Les critères utilisés sont généralement ceux de la NINCDS-ADRDA (McKhann, et al., 1984; McKhann, 2011). Ces critères sont globalement les suivants : 1) Présence d'un trouble de mémoire épisodique sévère (plus de 2 é.-t. sous les normes aux tests de mémoire standardisés); Présence d'une autre atteinte cognitive (plus de 2 é.-t. sous les normes aux tests standardisés); 3) Diminution de l'indépendance fonctionnelle. Les critères d'exclusion seront les mêmes que ceux de l'autre groupe.

Afin de pouvoir comparer les trois groupes, le MOCA et les tests exécutifs seront également administrés aux groupes TCL et DTA. Les données cliniques, neuropsychologiques et fonctionnelles ayant conduit au diagnostic de TCL ou de la DTA seront tirées des dossiers médicaux à la suite du consentement des participants. Il est ainsi possible que les tests exécutifs

et le MOCA aient déjà été administrés. L'obtention de ces données évitera donc d'avoir à retester inutilement les participants.