THÈSE
Pour obtenir le grade de
Docteur

Délivré par l'**Université de Montpellier**

Préparée au sein de l'école doctorale **I2S***
Et de l'unité de recherche **UMR 5506**

Spécialité : **Informatique**

Présentée par **M. Romain Endelin**
romain@endelin.fr

Méchanisme d'Optimisation du Raisonnement pour l'Actimétrie : Application à l'Assistance Ambiante pour les Personnes Âgées

Soutenance prévue le 2 Juin 2016 devant le jury composé de :

M. Mounir MOKHTARI	Professeur	Institut Mines-Télécom, Résident Univ. Montpellier, Associé LIRMM	Directeur de thèse
M. Vincent CHARVILLAT	Professeur	Univ. Toulouse	Rapporteur
Mr François BRÉMOND	Dir. Recherche	INRIA	Rapporteur
Mme Gabriele ANDERST-KOTSIS	Professeur	Johannes Kepler Universität Linz	Examinateur
M. Hamdi ALOULOU	Docteur	LIRMM, Institut Mines-Télécom	Examinateur
M. Philippe FRAISSE	Professeur	Univ. Montpellier	Examinateur

[Je considère que l'ingénierie de la connaissance sera une pierre fondatrice à l'analyse de modèles complexes, et permettra demain une révolution : le "knowledge science", en complément au "data science".]

* **I2S** : École doctorale Information Structures Systèmes

Thesis

To obtain the degree of Doctor

Delivered by the **University of Montpellier**

Prepared in the doctoral school **I2S***
And the research unit **UMR 5506**

Speciality: **Computer Sciences**

Presented by **Mr Romain Endelin**
romain@endelin.fr

Mechanism for Optimizing the Reasoning for Activity Recognition: Application for Ambient Assisted Living for Elderly People

Should be defended on June, 2nd 2016 in front of a jury composed of:

Mr Mounir Mokhtari	Professor	Institut Mines-Télécom, Located in University of Montpellier, Associated to LIRMM	Thesis director
Mr Vincent Charvillat	Professor	University of Toulouse	Reviewer
Mr François Brémond	Senior Researcher	INRIA	Reviewer
Mrs Gabriele Anderst-Kotsis	Professor	Johannes Kepler Universität Linz	Examiner
Mr Hamdi Aloulou	PhD	LIRMM, Institut Mines-Télécom	Examiner
Mr Philippe Fraisse	Professor	University of Montpellier	Examiner

[I believe that knowledge engineering will be the keystone for complex model analysis, and will create a revolution that will be known tomorrow as knowledge science, in complement to data science.]

* **I2S:** École doctorale Information Structures Systèmes

If your problem has a solution then...why worry about it?
If your problem does not have a solution then...why worry about it?

Dedicated to my parents, my wife and my daughter.

Abstract

Ambient Assisted Living is a promising research area. It aims to use information technologies to assist dependent elderly people on their daily life. The impact of these technologies could be dramatic for millions of elderly people and for their caregivers. This research area has developed consistently over the past few years, although slower than most other applications of the Internet of Things. This is caused by the inner complexity of Ambient Assisted Living. Indeed, Ambient Assisted Living requires a dynamic understanding of the context, as well as the disposal of numerous communication media in the environment surrounding the end-user. More precisely, researchers face difficulties in recognizing end-users' activities, as we can observe in the literature.

My research team have deployed our system in several environments, of which the most recent includes a nursing home and three houses in France. We adopt a user-centric approach, where end-users describe what they expect, and share with us their feedbacks and advices about our system. This approach guided me to identify Activity Recognition as a critical challenge that needs to be addressed for the usability and acceptability of Ambient Assisted Living (AAL) solutions.

Thus, the guiding line of this thesis work emerges naturally from the challenges we encountered during our deployments.

In the beginning of this thesis, I have been facing the practical problem of putting into place an actual deployment of our system. In this document, I describe the needs that emerged from our own observations and from the users feedbacks, as well from as the technical problems we encountered. For each of these problems and needs, I describe the solution we have selected and implemented.

From our deployments, my team and I were able to collect a large amount of operating data. I have created a platform to analyze Ambient Assisted Living data, also to allow rapid prototyping for Activity Recognition. By using this platform, I have observed problems with Activity Recognition, which is too often misleading and inaccurate.

A first observation is that the sensor events are sometimes disturbed by multiuser situations, when several persons are active in the home. Activity Recognition in these conditions is extremely difficult, and during this thesis my scope is solely focused on detecting multiuser situations, not recognizing activities in such situations.

I then seek to improve the quality of our reasoning engine. To do so, I have looked more precisely at some incorrect reasoning. I observed that the errors in reasoning come from the fact that the reasoner tries to be too precise or that, conversely, it infers too imprecise activities. I therefore propose a method to optimize the reasoning engine, so that it concludes with the best possible activity among several possible activities, by choosing the one that offers the best compromise between Precision and the risk of Inaccuracy in Activity Recognition. It should be noted that this contribution is independent of the method used for Activity Recognition, and can work with any type of reasoning.

I have formalized the concept of Accuracy, and provided a method to measure the Accuracy of a reasoning engine. This requires first to observe a ground-truth on the activity being performed.

This contribution brought me to introduce a hierarchical model for activities. Indeed, by applying the method described above on a hierarchical model of activities, the reasoning engine can be calibrated automatically to choose how precise it should be at recognizing an activity.

It goes without saying that these contributions are formally validated through this dissertation.

These contributions also lead to interesting perspectives, two of which are presented in this document, with proof-of-concept implementations.

Keywords

Inference Engine, Rule Factory, Ambient Assisted Living, Elderly People With Mild Dementia, Ambient Intelligence, Internet of Things, Context-Awareness, Activity Recognition, Semantic Web, Real World Deployments, Accuracy Measurement, Hierarchical Activities, Pervasive Computing.

Résumé

L'Assistance Ambiante est un domaine de recherche prometteur qui vise à utiliser les technologies de l'information pour venir en aide aux personnes dépendantes durant leur vie quotidienne. L'impact de ces recherches pourrait être déterminant pour de nombreux séniors ainsi que pour leurs proches. Cette discipline s'est développée régulièrement au cours des dernières années, mais tout de même plus lentement que la plupart des autres applications de l'internet des objets. Cela est dû à la complexité inhérente à l'Assistance Ambiante, qui nécessite une compréhension dynamique du contexte, ainsi que la mise en place de nombreux média de communication dans le lieu de vie de l'utilisateur. Plus précisément, les chercheurs rencontrent des difficultés avec l'étape déterminante de la reconnaissance d'activité, comme nous le montre la littérature.

Mon équipe de recherche a déployé notre système dans plusieurs environnements, le plus récent consistant en une maison de retraite et trois maisons individuelles en France. Nous adoptons une approche centrée sur l'utilisateur, où l'utilisateur final définit ce qu'il attend, et nous fournit des réactions et conseils sur notre système. De cette manière, nous pouvons apprendre de nos déploiements, et obtenir des informations pour répondre aux défis de l'Assistance Ambiante, y compris la reconnaissance d'activité.

Ainsi, la ligne directrice de cette thèse émerge des défis que nous avons rencontré durant nos déploiements. Au commencement de cette thèse, j'ai été confronté à la problématique concrète d'un déploiement réel de notre système. Je relate donc les besoins que nous avons vu émerger, par nos propres observations et par les retours des utilisateurs, ainsi que les problèmes techniques que nous avons rencontrés. Pour chacun de ces problèmes et besoins, je décris la solution que nous avons retenu et implémenté.

Une fois le système installé, mon équipe et moi-même avons pu récolter de nombreuses données sur son fonctionnement. J'ai tout d'abord mis en place une plate-forme d'analyse de données en Assistance Ambiante, permettant un prototypage rapide lié à la reconnaissance d'activité. En tirant profit de cette plate-forme, j'ai observé le problème posé par la reconnaissance d'activité, qui est une étape critique, mais trop souvent inexacte dans ses conclusions.

Pour faire face aux erreurs dans le raisonnement, je formalise la notion d'exactitude pour la reconnaissance d'activité, et fournit une méthode pour mesurer l'exactitude de notre moteur de raisonnement. Cela requiert d'abord d'observer une vérité terrain sur l'activité en cours, ou à défaut une estimation sur cette activité, d'une source autre que le raisonneur lui-même.

Je cherche ensuite à améliorer la qualité de notre moteur de raisonnement. Pour y parvenir, je m'attache à regarder plus précisément certains raisonnements incorrects. J'y observe que les erreurs de raisonnement viennent parfois du fait que le raisonneur essaie d'être trop précis, ou qu'à l'inverse, il est parfois trop imprécis dans les activités qu'il infère. Je propose donc une méthode pour optimiser le moteur de raisonnement, de telle manière

à ce qu'il conclue de la meilleure façon possible parmi plusieurs activités suspectées, en choisissant l'activité qui offre le meilleur compromis entre sa Précision et le risque d'Inexactitude de la part du moteur de raisonnement.

Cette contribution me mène à introduire une hiérarchie entre les activités. En effet, en appliquant la méthode précédente sur un modèle hiérarchique d'activités, le raisonneur est calibré automatiquement, pour choisir à quel niveau de précision il pourra reconnaître une activité.

Il va de soi que ces travaux sont validés formellement au sein de cette dissertation.

Ces contributions mènent aussi à l'émergence de perspectives intéressantes, dont deux sont présentées dans ce document et implémentées en tant que preuve de concept.

Mots-Clés

Moteur d'Inférence, Génération de Règles de Raisonnement, Assistance Ambiante, Personnes Âgées avec Démence, Intelligence Ambiante, Internet des Objets, Compréhension du Contexte, Reconnaissance d'Activités, Web Sémantique, Déploiements Réels, Mesure d'Exactitude, Activités Hiérarchique, Informatique Pervasive.

Author's Publications

Journal

[J1] Hamdi Aloulou, Mounir Mokhtari, Thibaut Tiberghien, **Romain Endelin**, and Jit Biswas. "Uncertainty handling in semantic reasoning for accurate context understanding". In: *Knowledge-Based Systems* 77 (2015), pp. 16–28.

International Conference

[C1] **Romain Endelin**, Mounir Mokhtari, Bessam Abdulrazak, Hamdi Aloulou, Firas Kaddachi, and Joaquim Bellmunt. "Activity Recognition Enhancement Based on Groundtruth: Introducing Accuracy and Precision Metrics". In: *Thirteenth International Conference on Ubiquitous Intelligence and Computing (UIC 2016)*. IEEE, submitted, 2016.

[C2] Hamdi Aloulou, Mounir Mokhtari, Bessam Abdulrazak, **Romain Endelin**, Thibaut Tiberghien, and Joaquim Bellmunt. "Simplifying Installation and Maintenance of Ambient Intelligent Solutions Toward Large Scale Deployment". In: *Fourteenth International Conference on Smart Homes and Health Telematics (ICOST 2016)*. Springer, accepted, 2016.

[C3] Joaquim Bellmunt, Thibaut Tiberghien, Mounir Mokhtari, Hamdi Aloulou, and **Romain Endelin**. "Technical Challenges Towards an AAL Large Scale Deployment". In: *Thirteen International Conference on Smart Homes and Health Telematics (ICOST 2015)*. Springer International Publishing, 2015, pp. 3–14.

[C4] Jit Biswas, **Romain Endelin**, Clifton Phua, Aung Aung Phyo Wai, Andrei Tolstikov, Zhu Jiaqi, Thibaut Tiberghien, Hamdi Aloulou, Philip Yap Lin Kiat, and Mounir Mokhtari. "Activity Recognition in Assisted Living Facilities with Incremental, Approximate Ground Truth". In: *Thirteen International Conference on Smart Homes and Health Telematics (ICOST 2015)*. Springer International Publishing, 2015, pp. 103–115.

[C5] Mounir Mokhtari, **Romain Endelin**, Hamdi Aloulou, and Thibaut Tiberghien. "Measuring the impact of ICTs on the quality of life of ageing people with mild dementia". In: *Twelfth International Conference on Smart Homes and Health Telematics (ICOST 2014)*. Springer, 2014, pp. 103–109.

[C6] **Romain Endelin**, Hamdi Aloulou, Jos De Roo, Stéphane Renouard, Thibaut Tiberghien, and Mounir Mokhtari. "Implementation of Allen's interval logic with the semantic web". In: *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems (MEDES 2013)*. 2013, pp. 252–253.

[C7] **Romain Endelin**, Stéphane Renouard, Thibaut Tiberghien, Hamdi Aloulou, and Mounir Mokhtari. “Behavior recognition for elderly people in large-scale deployment”. In: *Eleventh International Conference on Smart Homes and Health Telematics (ICOST 2013)*. Springer, 2013, pp. 61–68.

Book Chapters

[BC1] Bruno Salgues and **Romain Endelin**. In: Bruno Salgues and Norbert Paquel. *Télésanté, espoir du monde rural? Des objectifs à la construction des usages.* Michel Houdiard Editeur, 2013. Chap. Les futures solutions technologiques.

[BC2] Bruno Salgues and **Romain Endelin**. In: Bruno Salgues and Norbert Paquel. *Télésanté, espoir du monde rural? Des objectifs à la construction des usages.* Michel Houdiard Editeur, 2013. Chap. Les composantes technologiques.

Acknowledgement

Before I give thanks to all the wonderful people who contributed to this thesis, my first and sincerest gratitude goes to God, for all the wonderful opportunities He has placed upon my life, and for the strength He gave me to catch these opportunities, despite my numerous mistakes.

I owe everything to my parents. Not only you have always strived to give me the best education you could and you have always remained patient to me. But even today as an adult, you have still supported me, and I would not have been able to complete this thesis without your help.

I also want to give sincere thanks for my wonderful wife, Diti, for your patience, your daily support, for the many sacrifices you have made to let me complete this thesis. I also thank my no-less-wonderful daughter, Inaya, for the smile you keep bringing to me everyday, for the kindness you constantly show as you grow up, and also for the happy father you have made me become.

My thesis director, Mounir Mokhtari, comes right next in my gratitude. Working with you has been a great time, and I have learned a lot from your vision and your experience. I realize later on, that most of the times we used to disagree on something, the following few months proved you right, thanks to your experience; this was a humbling and beautiful lesson for me.

I also want to thank Stéphane Renouard, who should have been my co-director. I know all the sacrifices you have made for me, by pure generosity, and I sincerely hope I will have the chance to return you the favor one day.

Bessam Abdulrazak, you also had a huge contribution to my thesis, by teaching me how to write, and what academic really is. You make very big efforts to help me, despite your busy schedule, and I am really grateful for that.

Next up are my partners in crime, Thibaut Tiberghien and Hamdi Aloulou. What to say here? We have shared not only a strong colleague bound, but even a real friendship throughout these years, starting with the internship. I have seen you both evolve; Hamdi you deserve to rise high in academia, and Thibaut you are becoming an epic programmer, and manager and entrepreneur. I keep a good memory of every moments spent with you guys, and I wish to thank you for your ever helpful tutoring (and we all know I am not always an easy person to tutor…). I have a special thought for all the arguments we have had with Thibaut, these were all a pleasure, and Linux is definitely superior to MacOS.

I also want to thank every other persons who have shared this thesis with me. Joaquim Bellmunt comes first; you are the next in line for graduating your PhD. We have shared good time, and I really wish you best for your next few years, your thesis will be great! Firas Kaddachi and Fabien Clavier, too bad you guys arrive just when I leave. But I have great hopes for you, you are gonna do things the right way. I also want to mention specifically Mickael Germain, my first intern, and a great person. Working with you has been a real pleasure. I hope you finally use Vim! Many thanks also to Anastasia Lieva: this thesis

would not have been the same without your great involvement during your internship. I also want to thank all the other interns in Singapore, namely Guillaume, Clément, Cyril, Yoan, Laure, Joao and Anne-Sophie, for their great work and their enthusiasm in the team. Jérémie, you are out of this category, as you are much more of a friend than an intern. Thank you for all the inspiring discussion we had and we keep having.

I wish to thank Brigitte Choquet, from Saint-Vincent-de-Paul Nursing Home, for the strong involvement you have consistently shown in this project, and for your understanding towards a research project that was far from perfect at the initial setup. Along with Brigitte Choquet are every persons from Saint-Vincent-de-Paul, starting with Alain Roulland for your warm support, and Pascale Jeanne for the strong involvement you have given to this projects. But I do not forget Ludivine, Fanny and Sebastien, who not only contributed to this project, but also into making the Nursing Home a warm and lively place through their constant enthusiasm.

Jos de Roo also has all my gratitude, for the kind help you gave me about the Semantic Web, but also for the inspiring model you are as a person. As you say, "keep up the great work", and thank you for amazing tool that is EYE.

Many thanks to the administrative persons in Institut Mines-Télécom. My thoughts go to Ghislaine Leclerch, Jossely N'Kunga, Sophie Millat and Elisabeth Delaunay. The four of you have been incredibly supportive and nice to me. I would like to give particular thanks to Ghislaine for your unlimited patience facing my lack of experience and maturity towards administrative procedures, early in my thesis. As for the LIRMM, my first thought spontaneously comes to Nicolas Serrurier, for your wonderful human qualities. You are a unique person, and the keystone of LIRMM through your daily enthusiasm and your willingness to help people. I also want to thank Guylaine Martinoty, Virginie Fèche and Elizabeth Greverie for your kindness, your dynamism and your support in every events that make LIRMM a warm and lively place.

Besides administrative persons have also been many helpful and inspiring professors in LIRMM, who all contributed in their own way to make my experience in this laboratory as enjoyable as I could wish. Among these professors, I have a particular thought to Frédéric Comby, Philippe Poignet, André Crosnier, Olivier Strauss, Francois Pierrot, Abderrahman Kheddar, Andrea Cherubini and Philippe Fraisse,

Outside of LIRMM, I want to thank several actors in the community of AAL, who — despite their busy schedules — had the kindness to share their visions and experiences with me: Jit Biswas, Sumi Helal, Jacques Demongeot.

Last but not least come all my friends in the laboratories, who accompanied me during the internship or the thesis, namely (in alphabetical order): Adel, Adrien, Alexandro, Antoine, Aymen, Benjamin C., Benjamin N., Clément, Coralie, Emmanuel, Fabien, Flavien, Francois, Guilhem, Guillaume, Guillerme, Hélène, Ibrahim, Jean, Julien, Joris, Joven, Kévin, Mariam, Marion, Mathilde, Miguel, Namrata, Olivier, Reda, Rémi, Sabrina, Silvain, Stéphane, Swan, Thomas, Yacine, Vincent…Some of you have already completed it, others are still on the way. In any case, I wish you best for whatever you are doing now, and it was a pleasure to be your friend and share great discussions over lunches.

I do not forget my other friends, outside of the lab, who all in their own way have kept me motivated. The list could go on and on…

Contents

Abstract i

Résumé iii

Author's Publications v

Acknowledgement vii

Contents xiv

I Introduction 1

1 The Impact Of Ambient Assisted Living For Elderly People 3
1.1 General Context: An Ageing Population . . . 3
1.1.1 Facts . . . 3
1.1.2 Medical Insights . . . 6
1.1.3 Opportunities For Dependent Elderly People . . . 7
1.1.4 Alternative Solution: Staying At Home, With The Help Of Caregivers 9
1.2 Disciplinary Context: Ambient Assisted Living . . . 9
1.2.1 Pervasive Computing . . . 10
1.2.2 Ambient Assisted Living . . . 13
1.2.3 Smart Homes . . . 15
1.2.4 Discussions . . . 17
1.3 Problematics Of The Thesis . . . 18
1.3.1 Activity Recognition . . . 18
1.4 Thesis Positioning . . . 19
1.4.1 Conceptual Background . . . 19
1.4.2 Contributions Of This Thesis . . . 20
1.5 Structure Of This Document . . . 21

II Technologies vs. User Needs 23

2 Technological State of the Art 25
2.1 The Internet Of Things In Healthcare . . . 25
2.1.1 The Internet Of Things . . . 25
2.1.2 Applications Of The Internet Of Things Towards Healthcare . . . 28
2.2 Technological Considerations . . . 32
2.2.1 Architecture . . . 32

2.2.2 Physical Layer 32
2.2.3 Software 36
2.3 Activity Recognition 36
2.3.1 Activities, Context and Situation 37
2.3.2 Methods Of Activity recognition 40
2.3.3 Abnormal Activity Detection 46
2.4 Challenges Of Activity Recognition 48
2.4.1 Complex Activities 48
2.4.2 Switching Towards Real Deployments 49
2.4.3 Uncertainty Of The Reasoning Process 50
2.5 Discussion 50

3 User Needs Investigation **53**
3.1 Survey – The lifestyle of the elders 53
3.1.1 Content Of The Questionnaires 53
3.2 Population Studied 54
3.2.1 Elderly People 54
3.2.2 Caregivers 54
3.3 Daily Life Of Elderly People 58
3.3.1 Main Activities Of Elderly People 58
3.3.2 Health Disorders 59
3.3.3 Daily Activities 61
3.3.4 Contact With The Outside And Security 65
3.4 Aids Received By Elderly People 67
3.5 Openings 71
3.5.1 Search For Parameters Of Importance 71
3.6 Synthesis — The Needs Of Elderly People 73
3.6.1 Expectations Of Elderly People 73
3.7 Analysing Differences Between Elderly People And Their Caregivers 73
3.8 Conclusion 74

III A Framework for Ambient Assisted Living **77**

4 Addressing Technical Challenges **79**
4.1 Introduction 79
4.1.1 Objective: Large Scale Deployment 79
4.2 Initial Situation 80
4.2.1 UbiSMART v2 Implementation 81
4.2.2 UbiSMART v2 — Use Case 81
4.3 Problematics 84
4.3.1 How To Set-Up A Smart-Home? 84
4.3.2 How To Connect The Smart-Home To The Server? 85
4.3.3 The framework Was Not Practically Designed For The End-Users . 86
4.3.4 Scaling Up UbiSMART For Multiple Deployments 86

4.3.5 Summary . . . 86
4.4 Contributions . . . 87
4.4.1 Setting Up A Smart-Home . . . 87
4.4.2 Making UbiSMART More User-Centric . . . 93
4.4.3 Seamlessly Extending To Multiple Deployments . . . 95
4.5 Discussion . . . 105
4.5.1 Outcomes . . . 105
4.5.2 Openings . . . 106
4.5.3 Limitations: Activity Recognition . . . 108

5 Prototyping Activity Recognition Reasoning Engine 109
5.1 Motivations . . . 109
5.1.1 Features . . . 109
5.1.2 Relations Between Knot And The UbiSMART Reasoner . . . 110
5.2 Technical Specifications . . . 111
5.2.1 Architecture . . . 112
5.3 Data Analysis . . . 114
5.3.1 Getting Data From Database . . . 115
5.3.2 Loading The Data Structure . . . 115
5.3.3 Plotting Data . . . 116
5.4 Reasoning Workflow . . . 117
5.4.1 The Reasoning Process . . . 117
5.4.2 Implementing The UbiSMART Reasoning . . . 121
5.5 Simulator . . . 122
5.5.1 The Objects . . . 122
5.5.2 Example . . . 124
5.5.3 Opening . . . 127
5.6 Discussions . . . 127
5.6.1 Limitations and Perspectives . . . 127

IV A Methodology for Activity Recognition 129

6 Location Tracking and Multiuser Detection in Real Condition 131
6.1 Problematic . . . 131
6.2 Related Works . . . 132
6.3 Detecting Multiuser Situations . . . 133
6.3.1 Initial Dataset . . . 133
6.3.2 Reformatting the Dataset . . . 133
6.3.3 Inferring Presence and Movement of the Resident in Rooms . . . 137
6.3.4 Detecting Multiuser Situations . . . 137
6.3.5 Final Result: Location Tracking . . . 137
6.4 Discussions . . . 141

7 A Method for Improving Activity Recognition Reasoning 143
7.1 Introduction . . . 143
7.1.1 My Experience with Ambient Assisted Living . . . 143
7.2 Precision and Accuracy Issues in Activity Recognition . . . 145
7.3 Measuring Accuracy of a Rule Engine . . . 149
7.3.1 Gathering Ground-Truth . . . 149
7.3.2 Measuring Accuracy of Activities . . . 152
7.3.3 Discussion . . . 152
7.4 Improving the Decision-Making Process By Introducing A Score Of Quality 153
7.4.1 Introducing Precision to the Reasoner . . . 153
7.4.2 Introducing Hierarchical Activities . . . 153
7.4.3 Measuring Performance of the Reasoning Engine . . . 154
7.4.4 Discussion . . . 155
7.5 Integration . . . 156
7.5.1 Update the UbiSMART Reasoner . . . 156
7.5.2 Run The UbiSMART Reasoner . . . 156
7.6 Discussion . . . 157
7.7 Perspectives . . . 157
7.7.1 Detecting Faulty Rules . . . 157
7.7.2 Enriching Ground-Truth By Using Simulators . . . 158
7.7.3 Enriching Ground-Truth By Relying On User Habits . . . 159
7.7.4 Optimizing the Ground-Truth Acquisition By Involving Caregivers 159
7.7.5 Enriching Ground-Truth From Services . . . 160

8 Detecting Inconsistencies in Rule-Based Reasoning 161
8.1 Introduction . . . 161
8.2 Dynamic Environment Definition . . . 161
8.2.1 Houses Factory . . . 162
8.2.2 Rules Factory . . . 164
8.3 Detecting Problematic Reasoning Rules . . . 166
8.3.1 Suggesting new sensors . . . 166
8.3.2 Finding Conflicts Between Rules . . . 167
8.4 Implementation . . . 169
8.5 Conclusion . . . 169
8.6 Perspectives . . . 169

V Validation 171

9 Validation 173
9.1 Validating UbiSMART As A framework For Ambient Assisted Living . . . 173
9.1.1 Deployment . . . 173
9.1.2 Ground-Truth Data Acquired . . . 175
9.1.3 Feedback from End-Users . . . 176
9.2 Validating Knot as a Prototyping Framework for Activity Recognition . . . 177

9.2.1 Validating the Simulator . . . 177
9.2.2 Validating the Reasoning Engine . . . 178
9.2.3 Validating the Data Visualisation Tools . . . 180
9.2.4 Discussion . . . 181
9.3 Validating Location Tracking and Multiuser Detection . . . 183
9.4 Validating the Improvements on Decision-Making . . . 184
9.4.1 Updating the Reasoning Engine . . . 184
9.4.2 Acquiring Data . . . 185
9.4.3 Running The Reasoner . . . 185
9.4.4 Introducing Validation Metrics . . . 185
9.4.5 Results . . . 186
9.5 Validating the Detection of Problematic Reasoning Rules . . . 188
9.5.1 Validating the Houses Factory . . . 188
9.5.2 Validating the Rules Factory . . . 190
9.5.3 Validating the Detection of Missing Sensors . . . 192
9.5.4 Validating the Detection of Conflicting Sensors . . . 193
9.6 Discussion . . . 193

VI Conclusion **195**

10 Conclusion **197**
10.1 General Approach Selected In This Thesis . . . 197
10.2 Contributions . . . 198
10.3 Results . . . 199
10.3.1 Limitations . . . 200
10.4 Perspectives . . . 200

VII Appendix **203**

A Questionnaires — The quality of life of elderly people **205**

B Location Detection in Real Condition — Implementation **213**
B.1 Implementation . . . 213

Bibliography **244**

List of Figures **245**

List of Tables **249**

List of Listings **251**

Glossary **253**

Acronyms **255**

Part I

Introduction

A child among elders converses with his ears.

1 The Impact Of Ambient Assisted Living For Elderly People

In most cultures around the world, elderly people are wildly respected and play a significant role.

However, as the life expectancy increases and the population gets older, elderly people are often put aside of the society. Primarily, many people consider them primarily from a medical point of view. Others will think of them solely as a potential resource, mentioning the silver economy. Although medical concern is indeed critical for elders, and although the silver economy is a real opportunity for the economy, one must primarily consider elderly people as persons, with a right to dignity.

Building on this last statement, our society must adapt to an ever growing population of elderly people, and provide them with the assistance they require.

1.1 General Context: An Ageing Population

1.1.1 Facts

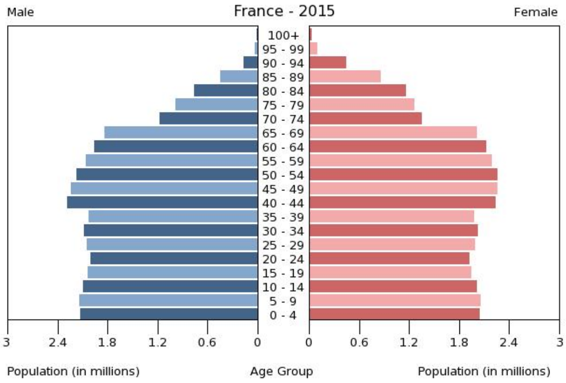
Ageing Of The Population

According to the French National Institute of Statistics and Economic Studies (Insee) [1] and the United States Census Bureau [2], in 2015 in France, 24.9% of the population is older than 60 years old, and 9.2% is older than 75 years old (Figure 1.1a). It is expected that in 2050, 31.4% of the French population will be over 60 years old, and 15.4% will be over 75 years old (Figure 1.1b). Conversely, 51% of the population is aged 20 to 59 years old in 2014, against only 45% expected in 2050.

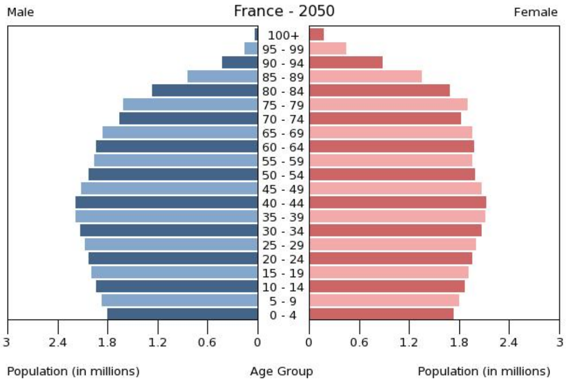
Similar trends can be found in many countries around the world, such as Singapore (Figures 1.1c and 1.1d) and the countries of the European Union (Figures 1.1e and 1.1f).

[1] http://www.insee.fr/fr/themes/document.asp?ref_id=T14F032

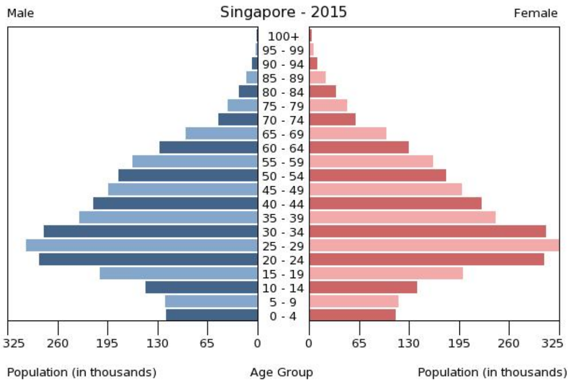
[2] http://www.census.gov/



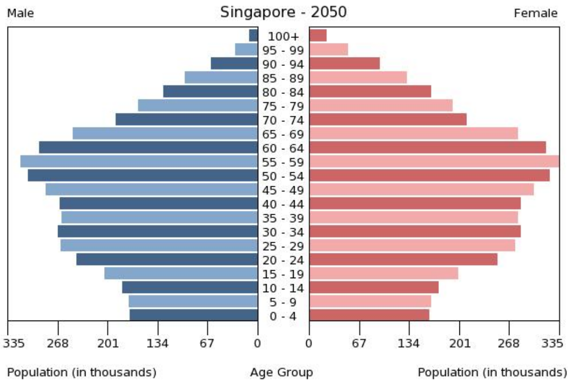
(a) France — 2015



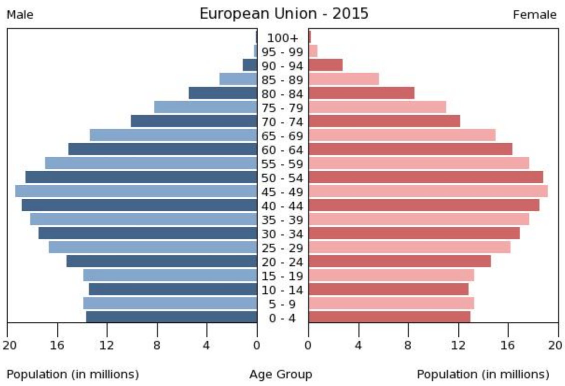
(b) France — 2050



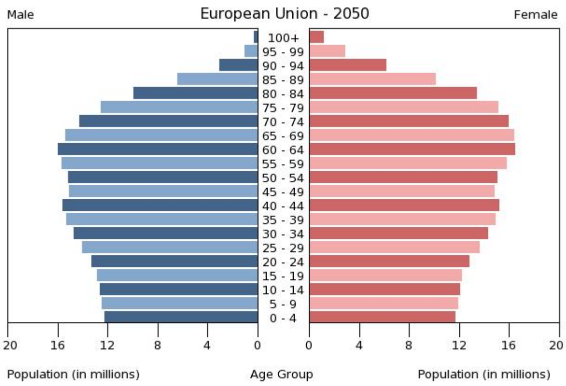
(c) Singapore — 2015



(d) Singapore — 2050



(e) European Union — 2015



(f) European Union — 2050

Figure 1.1: Population pyramids around the world

The Situation of Elderly People And Caregivers

To better understand the situation of elderly people, we have conducted a survey in 2014 over 120 elders and their caregivers [1], with questions related to their situation, their life habits, the problems they experience, and their expectations from assisted services. Further insights about this survey are presented in Chapter 3.

Figure 1.2 from our survey shows us that 49% of elderly people are living alone in their residence (whether they are not married, divorced, or widow). Only 3% of the interrogated persons were living in nursing homes, or other community centres.

We have also observed that 81% of elderly people had their family caregiver living under a 10 kilometers distance (Figure 1.3a). On the other hand, 6% of them can be considered isolated, as their caregivers live at more than 50 kilometers far.

We also see in Figure 1.3b that there are two categories of caregivers: those who live with the elders (and spend the entire week with them), and those who do not, and help the elders when they can, that is usually between 0 and 30 hours per week.

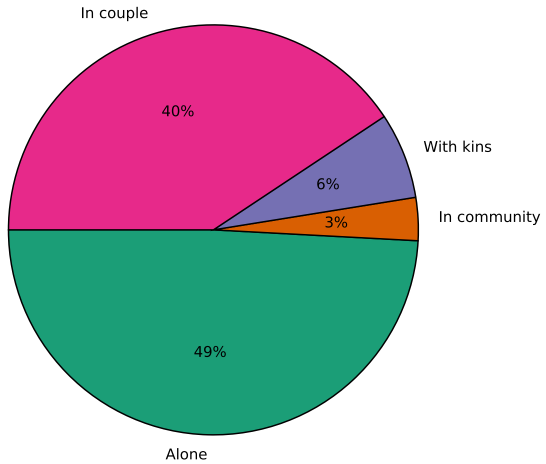
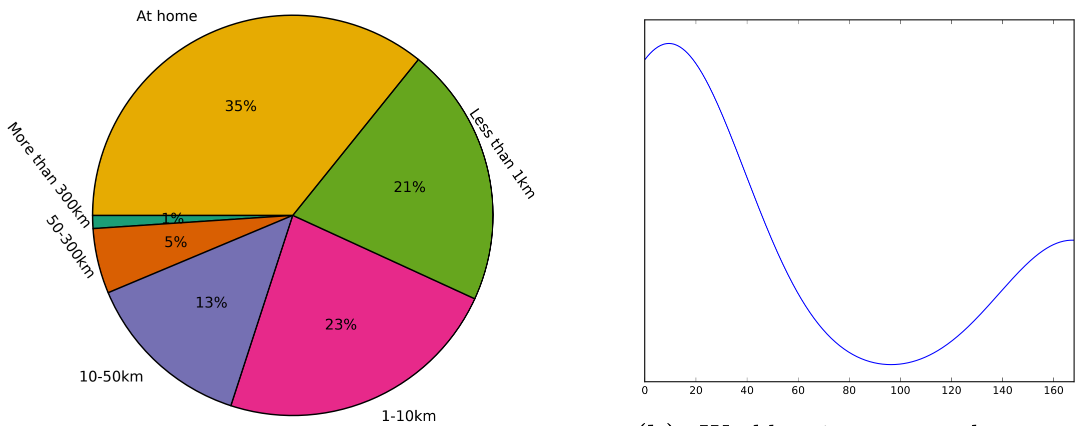


Figure 1.2: Persons living with elderly people



(a) Distance between elderly people and caregivers

(b) Weekly time spent by caregivers helping elderly people (hours)

Figure 1.3: The relation between elderly people and caregivers

1.1.2 Medical Insights

Following, I define what is ageing, from a medical point-of-view. I also extend the definition to pathological ageing and dementia, of which we are particularly interested in this thesis research.

Ageing

Ageing is an irreversible and inevitable process which increases the vulnerability and the number of deviations compared to ideal condition. In fact, nearly all the cognitive functions, the physical strength and the different senses show age-related decline. Among the changes experienced by ageing population, one can observe limitations in movement control with longer response time and less precise movements, spatial orientation problems, perceptual limitations concerning mainly the visual and auditory abilities and a decrease in processing speed. These changes appear at different rates for different people, and for different functions.

Based on the above presentation of ageing and related diseases, the process of ageing can be categorized in three main periods as follow:

Primary Ageing This concerns mainly people aged between 65 and 74 years old. It is also called normal ageing and refers to the inevitable biological and disease free changes that are typical among most people as they get older, especially gradual cognitive and physiological deterioration.

Secondary Ageing Also known as pathological ageing, it concerns developmental changes related to lifestyle factors and diseases, such as cardiovascular disease or dementia, for a population often aged between 75 and 85 years old. Such diseases are generally more frequent with age, but are not universally part of the ageing process.

Tertiary Ageing It refers to a great increase in physical and cognitive deterioration with rapid loss in organs and behavioral systems in the months before death. It is not so much correlated with age, but rather with the approach of death.

Pathological Ageing

Pathological ageing, introduced earlier as secondary ageing, is defined as an unexpected decline of one or several conditions (cognitive, physiological, social) of a person with advancing age. It can come forward as an unusual acceleration of changes under the scope of normal ageing, or as the apparition of changes unrelated to normal ageing. Although all conditions can be affected by pathological ageing, a particular focus is commonly put on the cognitive condition.

Mild cognitive impairment characterises a cognitive impairment beyond that expected for the age and education of a person. It is a transitional stage between normal ageing and dementia, for which the impairment has no major effect on the performance of Activities of daily Living (ADL). It is mainly seen as a risk factor towards dementia but at this stage, a person's condition can remain stable or even improve.

Dementia is a syndrome that can be caused by a number of progressive illnesses that affect memory, thinking, behaviour and the ability to perform everyday activities [2]. Alzheimer's disease is the most common type of dementia, and other types include vascular dementia, dementia with Lewy bodies and frontotemporal dementia. The boundaries between the types are not clear, and a mixture of more than one type is common. Dementia mainly affects older people, although there is a growing awareness of cases that start before the age of 65. After age 65, the likelihood of developing dementia roughly doubles every five years.

Dementia And Alzheimer Disease

Dementia is characterized by a progressive deterioration of intellectual and functional abilities, typically over a period of 7 to 10 years [3] and is classified into 5 stages (3–7) according to the Global Deterioration Scale [4].

At stage 3, the symptoms can be subtle and the patient can live independently without assistance. At stages 4 and 5, independent living becomes an issue, and in more advanced stages of the disease, the situation becomes critical, especially with verbal communication problems (aphasia), difficulty in identifying objects and persons (agnosia), and high-level disorder in performing familiar and learned tasks (apraxia). The elder, therefore, needs continuous support by a caregiver because he or she can no longer perform everyday tasks.

In epidemiology, the prevalence is the proportion of a population found to have a condition. The prevalence of dementia was estimated to 35.6 million people in 2010 and this number is nearly doubling every 20 years [5]. A study using a stochastic, multistate model was used in conjunction with United Nations' worldwide population forecasts and data from epidemiological studies of the risks of Alzheimer's disease in order to forecast the global burden of Alzheimer's disease by 2050 [6]. This study predicted that by the year 2050, the worldwide prevalence of Alzheimer's would grow fourfold to 106.8 million. It is estimated that 48% of cases worldwide are in Asia, and that the percentage in Asia will grow to 59% by 2050. The World Alzheimer Report 2010 [2] which focuses on the global economic impact of dementia estimates that the worldwide cost of dementia is USD 604 billions each year.

1.1.3 Opportunities For Dependent Elderly People

Elderly People Living In Their Home

Independent living for elderly people implies a number or upsides and downsides. Through autonomous living, elderly people maintain their dignity, their self-esteem by feeling responsible, and receive more stimulation. However, while everyone usually appreciate some autonomy, the excess of it can lead to loneliness and depression. Additionally, a person living alone is more likely to have accidents, with sometimes dramatic consequences. In case of a fall, it is not unusual for elderly people to spend an entire night on the floor, harmed and unable to call for any help.

Paradoxically, whereas a large number of elderly people develop depression and anxiety by living at home, they will most often refuse to hear about nursing homes [7, 8]. The intervention of professional caregivers is a good solution to postpone the nursing home while minimizing the risks for the elder, and take care of the actions they are not able to do themselves. However, in terms of economy, the solution of professional caregivers is barely viable. Private health insurance and public services struggle to find an economical model that would integrate every customers' need. Elderly people themselves don't always have enough financial resources to pay for professional caregivers, nor do their relatives.

Elderly People Going To Nursing Home

Nursing homes suffer from a negative reputation in the public opinion and face many criticisms. The most profound criticism addressed to nursing homes is that they are "halfway houses between society and the cemetery" [9]. In western countries the roots of these criticisms can be found at their very origin, during the industrial revolution in the nineteenth century [10]. At this time, it was customary for the elderly people to be taken care by their kins. For those who did not have this chance, they were placed in the almshouses, that were generally despised by the population and labelled as "warehouses" for the insane and the invalid ones. Almshouses have later evolved into nursing homes, and the quality of their services have gradually improved over time in developed countries, up to an acceptable level. Still, the opinion largely remains.

Even in societies with strong traditions of ageing parents living with children, such as in Japan, traditional living arrangements are becoming less common [11]. Also, the role of nursing homes is twofold: it does indeed protect dependent elderly people, as a family could do, but it also provides heavy medical tools and support within the daily environment, for those who need it. Such medical tools can hardly be installed in a normal house, and a full-time medical support is not a practical solution for average-budget people. Therefore, even in these countries, we see an increasing number of elderly people living in nursing homes, as the number of dependent people increases.

As for the elderly people themselves, they generally refuse to be placed in nursing home. It is not unusual to see people entering nursing home after several years of slowly-evolving mild dependence, and suddenly getting a dramatic degradation of their health within only a few weeks [12]. It is well-known that many elderly people are placed in nursing home after they have an accident in their residence; their kins worry because of the accident, and they choose nursing home as the safest solution. Yet, in many of these situations, the elderly people still feel able to live in autonomy, and consider the accident as an isolated event. Thus they don't understand their kins' decision, and often feel betrayed. This is a negative start for their new life within the nursing home, which partly explains the health degradation mentioned above.

Yet, in France as in many countries, nursing homes have greatly improved their quality of service, thanks to better regulations and a cultural change that implies a good involvement of the caregivers as well as more concern from the family of the elderly people [13].

From an economical point of view, nursing homes are expensive. In France the monthly cost for hosting an elderly people in nursing home varies around €2400 per month, and the resident is charged €1400 per month [14]. It is worth mentioning that the average retirement income in France is €1300 [15].

1.1.4 Alternative Solution: Staying At Home, With The Help Of Caregivers

Throughout the ageing process, elderly people gradually lose their abilities to live independently, while they generally refuse to go to nursing home. In many cases, they are placed in nursing home not because they are fully dependent, but because certain aspects of life bring difficulties and risks if they stay in their residence.

It appears that living an independent life is often a better solution for elderly people as well as for society. However, as the dependence of the elderly people increases, the nursing home becomes a necessity. But the nursing homes are not a viable solution in terms of economy, and the transition is often too brutal for the persons. Thus, the transition towards nursing home should be postponed as much as possible, and made smoother for elderly people.

The work of caregivers greatly contributes to maintain elderly people at home, and to reduce the risks. Unfortunately, the help of caregiver does not come without difficulties. In addition to being costly (about €1000 per month charged to the resident in France [16]), the support of the caregivers is by nature incomplete, as they can only visit the elderly people a handful of hours in a day, leaving them vulnerable the rest of the time. Moreover, the work of caregiver — whether a family member of or a professional — is extremely hard, often associated with depression and large drop-out rate [17].

The work of Caregiver must be acknowledged, as well as their human involvement, because it largely improves elderly people's life. However, in any way, ageing comes with difficulties, from which suffer the elderly people as well as the caregivers. This is why we want to emphasis the role that technology can play on elderly people care, through the development of Ambient Assisted Living.

1.2 Disciplinary Context: Ambient Assisted Living

In Section 1.1, I have stated my interest in helping elderly people to maintain an autonomous life, for as long as possible. I have also mentioned the difficulties suffered by caregivers — whether they are members of the family or professionals — to assist dependent people.

In the meanwhile, a major revolution is taking place in the numeric ecosystem, that can be of great use for assisting elderly people. Today the most familiar aspect of this revolution is known as the Internet of Things (IoT). In fact, the IoT is just one component of a wider paradigm, named the pervasive computing.

1.2.1 Pervasive Computing

Pervasive computing — equally known as Ubiquitous computing has been first imagined by Mark Weiser in 1991 [18], who had defined it as “The age of calm technology, when technology recedes into the background of our lives”.

Today, we could define pervasive computing as a software engineering concept where a user interacts with existing computing devices in different forms, including laptops, tablets and everyday objects, such as glasses, bed, shower etc. Ubiquitous computing is considered by Weiser as the third wave in the history of computer [19]. Still according to Weiser, the first wave was mainframe machines shared by a lot of people, and the second wave was the personal computer.

This new paradigm is also known as ubiquitous computing, “everywhere”, the physical computer, haptic computer [3] and “things that think”. Pervasive computing is present in many research topics, including distributed computing, mobile computing, sensor networks, human-computer interaction and artificial intelligence [20].

A New Paradigm For Human-Machine Interactions

The Pervasive computing is a new paradigm for Human-Machine Interactions (HMI). It is opposed with the Desktop model.

In the Desktop model, one human is facing one machine. The communication is initiated by the user through input devices (keyboard, mouse). The user then receives a feedback from an output device (screen). In a Desktop, we find several patterns for user interface, of which the most famous is Window, Icon, Menu, Pointer (WIMP) [4], that is default in most personal computers. The user needs to discover the interface entry points, and perform his operations through these entry points. The WIMP interface have proved a good ergonomic for most users, as well as the hypertext model [5], or sometimes Command-Line Interface (CLI) [6] for more advanced users. But by nature, they all suffer from the same flaw: it is the user needs to adapt to the machine. Although that may sound like no more than a small annoyance for a majority of users, it may be crippling for users with physical or cognitive deficiencies.

On the other hand, with the pervasive computing, the user should not adapt to the machine, but it is rather the machine that will adapt to the user. Computer are no longer well-bounded machines but rather an interconnected network of geographically distributed machines, integrated into the environment, with various shapes and capabilities. The process remains event-driven, but events are triggered by passive sensors, rather than explicit input devices. Therefore, each machine can activate or not, providing multiple interaction mediums for the user, depending on his needs.

Today’s challenge is that no well-accepted interface for pervasive computing has emerged yet, as it is for the WIMP model in the Desktop model. The research for such

[3] www.worldhaptics.org/Mission

[4] https://en.wikipedia.org/wiki/WIMP_(computing)

[5] http://www.w3.org/WhatIs.html

[6] https://en.wikipedia.org/wiki/Command-line_interface

interface is underlying every research in pervasive computing, and involves ergonomic as well as technology.

Scenario

In his visionary paper, Weiser [18] mentioned an interesting use-case of pervasive computing, which is not so far from today's reality. He envisions the day of a woman, named Sal, in a world where pervasive computing has fully reached its potential. On morning, when Sal starts moving in bed, the house guesses that she will wake up soon, and prepares a coffee. Newspaper have been integrated with computer, so you can easily save articles for later reference. Window curtains are opened or closed depending on the neighbours activities, to maintain privacy. Cars have automatic parking facilities.

The scenario could go on and on, and as Weiser says, at the time he writes this paper in 1991, "to extrapolate from today's rudimentary fragments of embodied virtuality resembles an attempt to predict the publication of Finnegan's Wake after just having invented writing on clay tablets." Today, in 2016, we live the infancy of his vision, and we can envision more naturally some of the outcomes.

The Internet Of Things

The ubiquitous computing is built upon a network of objects, named the Internet of Things (IoT). It is important to notice that the "internet" of the IoT refers to an interconnected network of "things", and thus does not necessarily have to rely on the classic Internet Protocol (IP), although it often does. In the rest of this document, I will use the expression Internet of Things indistinctly, whether the network of object is based on the IP protocol or not.

The IoT is a natural evolution of electronic devices, as more and more objects gain network capabilities [21]. In 2008, the number of things connected to the Internet exceeded the number of people on Earth [7].

Each object in the paradigm of the IoT share at least one of these three essential attributes:

Sensors Sensors measure and monitor real world activities;

Connection The connection can be achieved through wired cable, WiFi, or any other wireless protocol. It can connect directly to the Internet, or to another device.

Controllers Controllers process, analyze data received from sensors, and take the decision as appropriate.

Use-Cases

Arguably the most notorious connected objects today are the smartphones, but I may also mention the emergence of smart-TV and smart watches. The IoT is also under active expansion in many niche markets, often inspired by the *maker* and *DIY* movements [8]. For

[7] `http://postscapes.com/infographic-cisco-internet-of-things`
[8] `https://en.wikipedia.org/wiki/Makers:_The_New_Industrial_Revolution`

instance, we can mention plants that can tweets, flood meters, connected scales, luggage monitoring, doorbell, smart lights, wifi-enabled kettle, or smart clothes with embedded sensors.

The IoT includes other areas of technology, including telemetry, telematics, M2M-communication (Machine to Machine), smart grids, intelligent transport systems and laptops.

Impact Of The Internet Of Things

Economic Impact

The Internet of Things is a market with strong growth in the technology sector [22]. IBM opened a new unit dedicated to the IoT, and to working in partnership with the leaders of this sector AT&T, ARM Holdings, and The Weather Company. Cisco assumes that the IoT will generate more than $ 1.4 billion in economic benefits for the next 10 years [23], because it will help companies to increase their efficiency and improve their products and services. Obviously, the popularity of the IoT starts to grow.

There are large mergers and acquisitions in this area. WAVECOME — a company specialized in developing solutions for GSM / GPRS / CDMA — was acquired by the company Sierra Wireless, a leading global devices Machine-to-Machine (M2M). Gemalto, a world leader in digital security acquired Cinterion. One could also mention the acquisition of Motorola M2M by Telit, global supplier of wireless technology.

Three main actors in the field of IoT, Samsung, Dell and Intel have come together to design an open source common approaches that can simplify the development of protocols, software and peripherals within the organization Open Interconnect Consortium [9] (Figure 1.4). The consortium also includes the company Atmel, Broadcom and Wind River. The main purpose of this organization is the development of interconnection standard components of IoT system among themselves, regardless of their manufacturers and objectives (Figure 1). The first task being considered is the creation of a smart home system: remote control, energy conservation, access to media contacts and digital security.

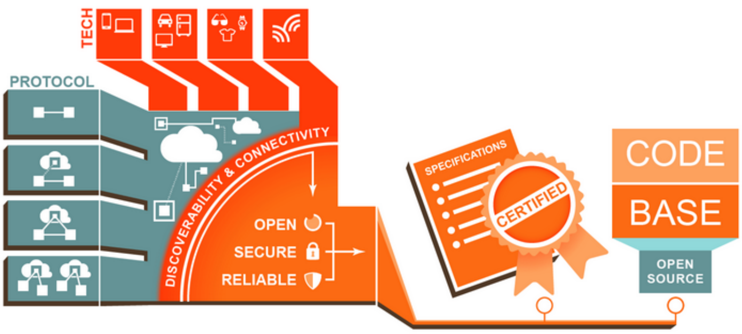


Figure 1.4: The concepts of the Open Interconnect Consortium Organisation

[9] http://openinterconnect.org/about/

According to the Open Interconnect Consortium, the requirements for the proper expansion of the IoT are:

Energetic efficiency For many remote objects, the battery charging is difficult, or even dangerous. Therefore, these objects should work as long as possible;

Profitability Objects with sensors must be widely distributed, so have a low production cost;

Reliability Sensors must be reliable, so that they are able to work in real environments, including sometimes outside.

11 % of French people are already equipped with a connected object, according to an Ifop poll conducted in November 2013 [10]. This is most often a smart bathroom scale, a watch or a bracelet. Another 12 % of the French plan to own such an object within 3 years. However the new technology adoption by older people is slower and less easy.

Acceptability Of The Internet Of Things For Elderly People

To better understand the perception of older people relative to the smart house a workshop with 30 leaders on law and services for elderly people was organized in 10 states in northeastern United States [24]. A variety of features have been identified as potential barriers, including ease of use, reliability, confidentiality:

1. Doubt on the reliability of the tools (50% of respondents);
2. The feeling of an intrusion in the daily (29%);
3. Distrust about the use of data (24%);
4. The fear of not knowing how to use the object (22%);
5. The fear of addiction, such as for smartphones (10%).

However, despite these barriers, participants generally appreciated the development and commercialization of these technologies.

1.2.2 Ambient Assisted Living

Among the multiple use cases of pervasive computing, the one use-case I am interested in is Ambient Intelligence (AmI), and particularly AAL.

[10] http://www.ifop.com/media/poll/2426-1-study_file.pdf

Ambient Intelligence

Ambient Intelligence has been defined as "a new vision of how people will interact with technology. Multiple, connected devices are embedded in the environment, adapting to, and anticipating our desires. It is above all, a human-centric vision of the future" [11].

AmI can be considered as layer built upon pervasive computing, to integrate some intelligence into an ubiquitous system. The intelligence takes the form of context-awareness, and also integrates some adaptive features to the system, where it knows how to react to the users need. A system of AmI must verify the following properties [25].:

Embedded Multiple networked devices are integrated into the environment;

Context-Aware These devices can recognize the user and his situational context;

Personalized The system can be tailored to the user's needs;

Adaptive It can change, according to the user's own evolution;

Anticipatory A AmI system can anticipate the user's desires, without explicit interaction.

Ambient Assisted Living — Definition

AAL is a subset of AmI, that applies the principles of AmI for assisted living. It aims to extend the duration when elderly people — possibly suffering from dementia or disabilities — can live in their home environment, by increasing their autonomy and assisting them in carrying out ADLs. The overall objective of AAL is to enhance the quality of life of elderly people. It can be integrated into many environments, such as a car, a street, or a house.

Use Cases

AAL has had multiple applications already, that will be listed more extensively in Chapter 2. Yet, I have found it interesting to include a few use cases in this section, so that the reader can visualize better the variety of what AAL can do.

AAL can be applied both to indoor and outdoor situations. One may think of a smart bracelet that would check for vital signs, a geo-location service through a GPS. Smart medicine box with reminders, smart kitchen to guide persons with diabetes, smart bed to detect sleep depravation are also defined as AAL applications. Serious-games and social services could also be integrated into AAL systems, by integrating data gathered with sensors as feed for the serious game, or by providing context-aware interfaces of social medias for cognitively deficient users.

[11] `http://www.ambient-intelligence.net`

A multidisciplinary Approach

AAL is a multidisciplinary approach to assisted living. It involves both technological and paramedical components. As for the paramedical, a good AAL system must be designed to support the end-user's well-being as well as his safety. It requires a special attention on his needs, on his capabilities, on ergonomic and on his medical condition. In terms of technological requirements, in my point of view, AAL systems are composed of multiple layers:

Hardware components Hardware components can be divided into three types:

- **Sensors** Sensors feed the system's knowledge of the environment
- **Processors** Processors maintain awareness of the context and take decisions
- **Actuators** Actuators provide a feedback to the user

Communication layer The communication layer spreads messages among multiple objects.

Software components I define the following software components for AAL:

- A reasoning engine for context-awareness
- A decision-making engine
- User-interfaces integrated into the actuators

1.2.3 Smart Homes

As we have stated in Section 1.1, our focus is on helping elderly people to stay in their home for as long as they can. Therefore, our use case for AAL leans towards the creation of a smart home [26]. Such smart home would assist elderly people in his daily life, advice them to drink water or take medicine, give them hints of what to do when they wake up, guide them back to their bed, etc.

Scenario

In his wonderful short-novel "There will come soft rains" — written in 1950 and published as a part of the "Martian Chronicles", Ray Bradbury describes an extremely modern house, that performs a variety of daily tasks to help and entertain its residents. However, in that story, the house is fully automated, but cannot be considered as smart. Indeed, the house still performs blindly its routine, and is not aware that its residents are already long-time dead.

On the opposite, a smart home should be aware of the resident's context, in order to provide him with the right service when he is facing a certain situation. If a person is watching television, and forgets to take his daily medicine, a message could appear on the television, advising the patient to take his medicines. Another use-case for a smart home is to forward the proper information to the proper recipient. For instance, such system would tell to a doctor that the elderly resident has being suffering from sleep depravation recently. It may as well alert a family caregiver when a fall is being suspected.

Throughout these examples, I have purposely excluded use cases whose interest is solely focused towards entertainment or multimedia services. Indeed, although these applications can also be defined as part of a smart home and AmI, they do not match the scope of AAL — except for the case of serious game.

Smart Homes — Requirements

As I have defined in 1.2.1, pervasive computing, and more specifically smart homes, are based a new paradigm of HMI, where it is the machine which adapts to the end-users. This brings new challenges in terms of conception, ergonomic and technology.

Among the parameters that need to be considered for the conception of a smart home are the cost of the system, the simplicity for the user, the security and privacy concerns, the granularity of the services and their relevance to the user's profile, the selection of the output devices, the choice and the granularity of the sensors, the adaptability of the system in different environments, and its integration in a wider scope, such as a smart-city:

Cost of the system: To really benefit to people, a smart home needs to be sold at a reasonable price. This excludes cutting-edge sensors and heavy construction work.

Simplicity for the user: AAL solutions are designed for users with cognitive or physical deficiencies, and most generally to elderly people who often are not familiar with modern computer systems. Thus, accordingly to the Pervasive Computing specifications, the smart home needs to offer intuitive interactions with the user, with no learning curve.

Security and privacy concerns: The purpose of a smart home is to guarantee the security of its resident. Through the use of sensor, AAL systems gather sensitive informations. I therefore emphasis the importance of a secure system, as it will be described in Chapter 4. Additionally, a faulty service can have critical consequence, (i.e., a false negative where the system does not detect an accident). Thus, a special attention must be given to the reliability of the reasoning engine.

Adaptability of the system: There are many kinds of residences, from one-room apartment to big houses. The end-user might be living totally alone or having regular visits. He might have very regular routines, or might not, etc. Facing this diversity, a smart home must be extensible to any environment, with a minimal payload when extending to a new environment. This adaptability has to be found both in the choice of sensors and services, as well as in the reasoning engine.

Choice of the services: An inclusion process with the elderly people, the family, the caregivers and the doctors is necessary, in order to determine clearly what are the needs of the elderly people, and what services could fulfill these needs.

Choice of output devices: Deeply present in the concept of pervasive computing is the importance of using the right device, depending on the context. If the end-user is deaf, it is meaningless to use speakers, nor does it when the end-user is in the kitchen and the speakers are in the living room. Similarly, a person cooking won't be able to use her hand to press a button, and an elderly people with dementia wouldn't like to hear a voice at night, coming out of nowhere, even if the intention is only to guide him to his bed.

Choice of the sensors: The first criteria to guide the choice of sensors in a smart home is the relevance of the data it will gather. This relevance has to be determined according to the services offered. The granularity of the gathered data must also be considered, as too imprecise sensors are meaningless, and too precise sensors are a waste of resources. One must also pay attention to the acceptability and privacy; CCTV and microphone are often rejected, for good reasons. Finally, several practical concerns, such as the wireless capabilities, the autonomy, the size, and even the beauty of the sensor (so that end-users can accept it) must also be taken into account.

Integration to a wider scope: A smart home can already be extremely helpful on its own, but its scope is limited to the house. In a long-term vision of smart-city, smart homes should be totally integrated within the city's information system. Entry points with other systems need to be defined, but on the other hand, some private data must remain within the context of a house. Thus a policy must be defined to share only the relevant data.

A proper reasoning engine: The underlying condition for a smart home to be effective is that it can perform a reliable activity recognition. The reasoning engine is the brain of our system, and every interactions with the end-user depends on the inferred activities. Even with all the above requirements satisfied, a poorly designed reasoning engine would fail in recognizing the situation, and invalidate any service. This problem is the major focus of my thesis work, and will be described extensively through this document.

1.2.4 Discussions

In the above section, I have defined the scope of a smart home, as an AAL system. The design of AAL system relies on the wider paradigm of pervasive computing.

Through our research, my research team and I integrate smart home functionalities for elderly people, in order to support them and their caregivers in their daily life, and help them living in their residence for as long as possible. However, we have seen that the creation of a smart home is non-trivial, because of its pervasive nature. The technology and ergonomic have to be considered accordingly, by creating a distributed architecture with sensors as input devices, and output devices integrated into the environment.

The most critical part of AAL for elderly people with cognitive or physical deficiencies is Activity Recognition. Even with the best technologies and a right choice of services, an

AAL system will perform poorly without a reliable Activity Recognition engine, and may even become dangerous.

Yet, in the context of a smart home, it is uneasy to create a reasoning engine that will perform Activity Recognition, as well as to evaluate the reliability of this reasoning engine.

1.3 Problematics Of The Thesis

This thesis work focuses on two different layers of AAL, that are the ability to perform a large-scale deployment of AAL system, and the reliability and flexibility of reasoning engines to perform Activity Recognition.

1.3.1 Activity Recognition

The most critical component of AAL — Activity Recognition — is also the most challenging one.

Activity Recognition is often generalized as context-awareness. Context has been defined by Dey [27] as "any information that can be used to characterise the situation of an entity", and therefore context-awareness is a global knowledge of the context. Activity Recognition is more specific, as it uses our awareness of the context to extract one particular information: the activity being performed. An activity is usually modelled as a sequence of atomic actions, executed together in order to complete a specific purpose.

In the context of AAL, context-awareness and Activity Recognition are necessary to trigger the right service on the right media, given a certain situation. The most critical case is to send an alert when a dangerous activity is detected, or even if the current activity is not recognized, and therefore considered as suspicious. But sometimes, the system is actually looking for the absence of a specific activity over a period of time (e.g., a reminder for the end-user to take medicine).

Overview Of The Literature For Activity Recognition

The literature for context-awareness is roughly divided into two approaches: data-driven approaches and knowledge-driven approaches [28].

Data-driven approaches use classical methods of machine-learning to perform activity recognition. They benefit from a well-studied discipline, with a large corpus and powerful algorithms. However, in the case of context-awareness, a classic Machine-Learning approach may show its limits. Indeed, context models are complex: they include temporal information in a dynamic environment, data gathered by the sensors is often incomplete. In the absence of a generalized model, each residence requires a new learning process to adapt to its specificities, and most often data are not labelled (due to the absence of ground-truth), reducing the applications to clustering algorithms, that perform poorly in such complex model.

On the other hand, knowledge-driven approaches are based on an expert knowledge of the environment. The expert knowledge will be formalized through a logical model, on which low-level knowledge — sensor data — will be injected. Then, a rule-based reasoning will be performed, to emerge high-level knowledge — context-awareness — from the sensor data. Knowledge-driven methods may be criticized as they are vulnerable to the *a-priori* knowledge of the expert. But, thanks to the abstraction capabilities of logical models, they often perform well on context-awareness for AAL, whereas data-driven approaches sometimes struggle with the complex nature of the context.

A more thorough analysis on the literature of context-awareness can be found in Section 2.3.

1.4 Thesis Positioning

My thesis is built upon the solid research work of Professor Mounir Mokhtari on AAL and Smart Home over the past 15 years [29], and the doctoral theses of Doctor Hamdi Aloulou [30] and Doctor Thibaut Tiberghien [31]. I have worked on it in parallel with the ongoing thesis of Joaquim Bellmunt Montoya.

During their doctoral theses, Aloulou & Tiberghien have developed the Ubiquitous Service MAnagement & Reasoning archiTecture (UbiSMART) framework for AAL, which I have been using and contributing during my own thesis work.

1.4.1 Conceptual Background

In this section, I describe our conceptual background, that is the choices my research team and myself have made, which position our work inside the community of AAL.

Using Ontologies For Our Reasoning

I have briefly described the literature for context-awareness in Section 1.3.1, showing that it is divided between data-driven and knowledge-driven approaches.

Looking at the pros and cons of these two approaches, our reasoning engine is based on a knowledge-driven approach. Technically speaking, we define an ontological model using the Semantic Web, Notation3 (N3) formalism [32] and OWL [33], with the Euler Rule Engine [34] to perform the reasoning.

However, a persistent issue with knowledge-based system, that is partially addressed during this thesis, is the certainty of the reasoning. Logical systems tend to consider every inference as certain, whereas in the case of context-awareness, uncertainty is to be found at many layers, from sensor data to the reasoning rules and the human behaviour.

Deployments In A Real Environment

To validate the quality and the relevance of our system, we have decided to deploy it on real environments, in addition to laboratory experiments or computer simulations. We do not reject the use of laboratory experiments and computer simulations — I use it too for this thesis —, but we believe they are not enough. AAL is a practical discipline, with a lot of qualitative aspects such as user needs, user acceptance and social implications of the system. Those aspects cannot be tackled by solely theorizing in a laboratory. Therefore, we emphasis real deployments, and we have deployed our system in a nursing home and in three elderly people's residence.

From these deployment we have gained feedbacks from the end-users, providing useful hints about ergonomic and in the choice of the services we offer. Additionally, through real deployment we have also been facing unsuspected issues, that would not have been found in a laboratory, such as problem with the wireless networks (or even an absence of Internet network at elderly people's houses). These feedbacks and issues will be described later on in this document.

We can only encourage other researchers in the field of AAL to extend their experiments beyond laboratories and perform real deployments, as much as possible.

Transferring The Complexity From Hardware To Software

We have mentioned in Section 1.2.3 that a smart home should be offered at a reasonable price, with a minimal installation cost, and should be adaptable to multiple environments. We have therefore decided to limit the complexity of the deployment, by using simple sensor as much as we can, and to transfer the complexity towards the software and the reasoning. The choice of the sensors will be discussed later on in this document, as well as their integration with the reasoning engine.

1.4.2 Contributions Of This Thesis

Validation And Improvement Of Reasoning For Activity Recognition

In the beginning of this thesis, I have been facing the practical problem of an actual deployment of our system. From this deployment, I have observed that Activity Recognition is too often inaccurate.

A first explanation is that the sensor events are sometimes disturbed when several persons are active in the residence. I have therefore defined an algorithm to detect these situations, in order to pause the Activity Recognition when the resident is not alone.

I also formalize the concept of accuracy for Activity Recognition, and provide a method to measure the accuracy of a reasoning engine. This method requires first to observe a ground-truth on the current activity, or at least an estimate of the activity, from a source other than the reasoner itself.

I then propose a method to optimize the reasoning engine, so that it concludes in the best possible activity among several possible activities, by choosing the one that offers the best compromise between precision and the risk of inaccuracy from the reasoning engine. It should be noted that this contribution is independent of the reasoning engine used, and can work with any type of reasoning.

This contribution also brought me to introduce a hierarchical model for activities. Indeed, by applying the method described above on a hierarchical model of activities, the reasoning can be calibrated automatically to choose at which level of precision it can recognize an activity.

Additionally, I have also developed two algorithms for assisting AAL experts in designing Activity Recognition reasoning engines. These algorithms detect faulty reasoning rules, conflicts between rules, and even suggest the deployment of specific sensors to improve the quality of the reasoning.

Technical Contributions

Through our deployment, I have seen the emergence of numerous technical requirements, several of these have been addressed in this thesis. In this thesis document, I will focus on the following contributions:

– The creation of a home gateway, that plays as a bridge between the house deployed and the server hosting our system. This includes a large work on sensor integration, reliability, automation, and ease of maintenance.

– The creation of a web-based production-ready AAL framework, which handles reasoning engine and service provisioning.

– A framework for rapid prototyping reasoning engines for Activity Recognition. This framework also includes powerful data analysis tools to emerge insights from the data obtained in deployments.

These technical contributions, partly developed in collaboration with Aloulou, Tiberghien and Bellmunt, are contributions to the discipline of AAL, and innovative thanks to the focus we have given to real deployment rather than laboratory work.

1.5 Structure Of This Document

Chapter 2 reviews the literature work on AAL, to give a wider context and better position my contributions. Chapter 3 summarizes an analysis we have performed with 120 elderly people and with their caregivers, in order to understand the needs of the users and address it the best possible way. Chapter 4 describes the technical challenges we have been facing during our deployment, and the solutions developed to respond to these challenges. Chapter 5 is a necessary foundation towards any consistent improvement of a reasoning engine for activity recognition. It describes the framework I have developed for data analysis and rapid prototyping of activity recognition. Chapter 6 describes a method for detecting several persons acting in the house, from sensor events. Chapter 7 is arguably the keystone of this thesis document. It describes a formal method to calibrate a reasoning engine for recognizing activities with an optimal precision and accuracy, based on real observations of the environment. Chapter 8 introduces an abstraction layer on top of ontological models, and leverages it to detect inconsistencies in a rule engine. Chapter 9 validates the contributions described in this thesis document. Chapter 10 concludes on this dissertation.

Part II

Technologies vs. User Needs

A people without the knowledge of their past history, origin and culture is like a tree without roots.

Marcus Garvey

2 Technological State of the Art

This chapter discusses the process of development and evaluation of AmI infrastructures, with a focus on context-awareness, tackling the ongoing problems of Activity Recognition. Section 2.1 describes the evolution of the IoT and its applications for healthcare. Section 2.2 mentions the technical challenges that appear with the IoT and with smart homes, and lists several implementations. Section 2.3 focuses on the challenges of context-awareness and Activity Recognition. Section 2.4 emphasizes on the current deadlocks of context-awareness, that are the gathering of data and ground-truth, and the validation of our Activity Recognition. Finally, through a discussion in Section 2.5, I explain the choices that I have done on the AAL platform — UbiSmart, and more particularly how I position my thesis work towards the state of the art.

Over the next 20 years, the number of older adults over age 65 is expected to double [35]. The majority of people aged 75 years or more become frail to the point of need for assistance. People living longer than ever, more medical care is needed, which generates higher costs for healthcare. 42 % of people who reach the age of sixty years will end their life in a retirement home [36]. The 58 % remaining can not or will not pay such services, and prefer to live in their own home, sometimes despite the presence of chronic diseases and medical dependence. The IoT can play a major role in this area [37].

2.1 The Internet Of Things In Healthcare

2.1.1 The Internet Of Things

An intelligent environment is one that collects data on residents to fit their needs and achieve the safety objectives, safety, efficiency and comfort. Currently there are a lot of AmI systems. These systems vary from the least expensive (easily integrated into existing automation systems) to complex industrial solutions. An ambient intelligence system can be identified by certain characteristics [38]:

Context-Awareness Takes advantage of the contextual and situational information;

Customization Adapted to the needs of each individual;

Anticipation Anticipates the needs of a person without conscious mediation of the individual;

Adaptation Adapts to the changing needs of individuals;

Ubiquitous Integrated in our everyday environments;

Transparency Are deep in our daily lives in a discreet way.

On to Business Insider, sales of smart homes will reach $ 61 billions this year and up to $ 490 billions in next 4 years [39] Google has started to work in this area in 2014 with the acquisition for $ 3.2 billion of the thermostat and alarm maker Nest. Then Google acquired Dropcam, the specialist of intelligent home security, and the developer of the revolver network devices. As a result of these acquisitions, Google has signed contracts with leading device manufacturers and suppliers of home automation products.

SmartThings, has built its business model around using simple sensors, monitors and applications that allow users to control all home via a smartphone. This allowed the company to earn $ 3 million in 2014.

The MIT collection of integrated devices project, called E21s, creates intelligent spaces in offices, homes and vehicles. E21s provide large amounts of embedded computing and interfaces for camera and microphone. Users naturally communicate in the spaces created by the E21s, without being aware of particular interaction.

Boston University researchers have created an Independent Living Assessment Tool, sponsored by Phillips, which offers three areas of intelligent assistance: moving safely in the intelligent environment, managing the activities of daily living, and make regular tasks [1].

The Start-up Wig-Wag [2] offers an infrastructure that is compatible with many existing smart devices, and open to modifications and ordinary users and developers. Wigwag creates “rules” — similar to IFTTT [3] recipes — to control any device. These can be Wigwag sensors and actuators, websites, Raspberry Pi services, Arduino, Belkin WeMo, Philips Hue etc. The creation of the rules can be done not only through the graphical user interface, but also in JavaScript, via the DeviceJS environment that Wigwag developers positioned as a way to write applications for the smart home in the same way as Web applications.

The ZigBee Alliance [4] and Thread Group [5] have announced a partnership to enable the bookseller ZigBee Cluster run on the networks of Thread. ZigBee Alliance is an association of companies with the objective of allowing reliable tracking, cost effective and powerful wireless networking products. ZigBee is a wireless network standard used for communication between a variety of everyday devices to ensure the comfort and safety

[1] `http://www.lifelinesys.com/content/`
[2] `http://store.wigwag.com`
[3] `http://ifttt.com`
[4] `http://www.ZigBee.org`
[5] `http://www.threadgroup.org`

of environments. Thread is a wireless networking protocol that supports multiple IP protocols with low bandwidth, and provides developers mesh networks with low energy consumption, direct Internet access and Cloud for each device. Both organizations remain independent, but working together, they can provide an interoperable solution to help streamline the development of connected objects and ultimately improve the experience of smart home residents.

Amazon also offers the smart home service Amazon Echo [6], with a first stage of integration with Belkin WeMo and lights Philips Hue. However, it seems that currently, Amazon Echo only supports WiFi, which can limit the options and speed of response to connect certain products of the smart home.

Quirky [40], founded in 2009, calls the Internet community collaborators that come with ideas for connected products. In partnership with General Electric, Quirky has created a line of objects connected by the names of both companies. Quirky sees the smart home as part of a broader IoT, and tries to focus on objects that are affordable to the public. And as every clients have not fully embraced the concept of the IoT, Quirky educates by a serie of advertisements on the principles of the IoT.

Several efforts were made by researchers to design non-marketed systems, sometimes in open-source.

The system ThinkingHome [7] can manage scenarios at home, as they assign an action to some events in the house, ie. trigger the detector, press the button or receive a message on Twitter. When an event occurs, the scripts to control home appliances are executed. The web interface is accessible from any device (PC, tablet or smart phone) connected to the network. It displays the information one needs in daily life (bus schedules or weather forecasts), and also provides the ability to remotely control home or change system settings. If the system lacks certain features or there is a need to work with non-standard devices (eg, at the base of different controllers as Arduino [8]) — it is possible for the user to write and share his own plugin.

The home automation system Majordomo [9] (Major Domestic Module) is a free software platform for integrated automation and IT support for life. This system can be installed on any operating system and is lightweight in terms of energy and memory. It can be used for the organization of the environment of the personal information. This project is part of the smartliving ecosystem [10].

[6] `http://www.amazon.com/oc/echo/`
[7] `http://thinking-home.ru/system`, in Russian
[8] `http://www.arduino.cc/`
[9] `http://majordomohome.com`
[10] `http://smartliving.ru` (in Russian)

2.1.2 Applications Of The Internet Of Things Towards Healthcare

Range Of applications

The possibility of deployment of an ambient environment means that residents and other interested users will receive access to all the deployed services, at any time and at any place, depending on their privileges towards the services.

The potential range of services is very broad and covers all intelligent support services that can make everyday life easier: Kleinberger et al. classifies these services in three categories, as shown in Figure 2.1 [41]. The more critical the service, the more intelligence it tends to require from the system. So the system must interact with a person in a multi-modular way, and ideally should enable natural communication, as if the person would interact with another person and not with an artificial system.

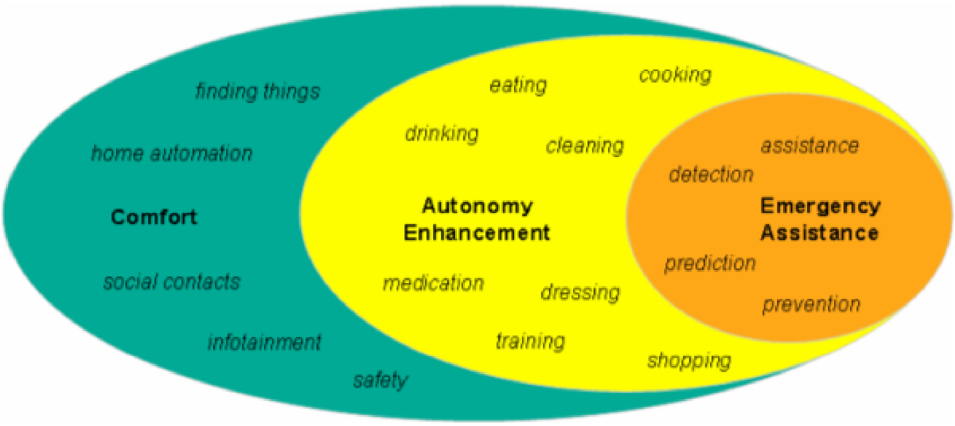


Figure 2.1: Ambient Intelligence Services

By detecting problems, smart environments can actively assist in prevention, for example by helping or suggesting actions in critical situations such as loss of consciousness, fall, injury etc. An escalating of service can help to reduce the criticality of false negative and false positive alarms: if after early stages of proactive support (the instructions for the person, family integration in decision making, etc.) the system can not prevent the critical situation, it contacts the emergency medical service and provides information on the current status of the assisted person. In [42], Mourouzis et al. propose an adaptive user interface platform, for personalized user experiences.

Based on [38], Table 2.1 describe several applications of AmI for healthcare. We observe that AAL is focused on environmental sensors, more than body sensors.

Industrial Applications

Health professionals may see in AmI an innovative solution for communications, as well as transparent and unobtrusive monitoring tools.

Class of Applications	**Goals**	**Environmental Sensors**	**Body Sensors**	**Methodologies**
Continuous Health Monitoring	Using sensor networks for monitoring physiological measures (ECG, EEG, etc.)	✓	❍	Actvity Recognition
Continuous Behavioural Monitoring	Using sensor networks for monitoring human behaviours (Watching TV, Sitting, etc.)	✓	✓	Activity Recognition
Monitoring for Emergency Detection	Using sensor networks for detecting hazards, falls, etc.	❍	✓	Activity Recognition
Ambient Assisted Living	Creating smart environments for supporting patients and elderly during their daily habits	✓	✗	Activity Recognition, Decision Support
Therapy and Rehabilitation	Supporting people who require rehabilitation services with remote and autonomous systems	❍	✓	Activity Recognition, Decision Support
Persuasive Well-Being	Systems aimed at changing persons attitudes in order to motivate them to live a healthier life style	✓	✗	Activity Recognition, Decision Support
Emotional Well-Being	Ubiquitous systems based on neurological and psychological insights to analyze emotions and imprve well-being	✓	✓	Activity Recognition
Smart Hospitals	Improving communicatio among hospitals stakeholders through ubiquitous technology	✓	✗	Decision Support

Table 2.1: Applications of Ambient Intelligence for Healthcare
Mandatory: ✓, Optional: ❍, Not required: ✗

SACHA [43] is a French project: the creation of a patch whose sensors can geotag patients with dementia (Alzheimer, Parkinson). The patch emits a warning signal in case of a fall or accidental fugue and thus enables to rescue the person as quickly as possible.

The E-Celcius[44] French project is a capsule to swallow, that can measure and transmit every 30 seconds the body temperature via radio frequency technology to prevent postoperative infections.

In Britain the development and deployment of smart homes were made in the Bath Institute for Medical Engineering (BIME) in the draft framework Gloucester Smart House [45]. The project was carried out with Housing21 and other technical partners to the creation of a smart apartment for people suffering from dementia. This required good coordination and communication with several agencies from electricity supplier to the ethical and social services.

Another system, COACH provides guidelines to continue a task for patients suffering from Alzheimer [46]. It uses a detailed representation of the process' steps for hand-washing, and is based on visual recognition techniques. If the user is unable to complete a particular steps, he receives detailed instructions.

The system of Pollack et al. [47] creates reminders on daily activities by reasoning on possible disparities between what the user is supposed to do and what he actually does.

Google bought the US start-up Lift labs [11] which produces spoons for people with Parkinson disease. When the sensors detect the shaking of the hand of its user, the spoon responds immediately by neutralizing to avoid spilling its contents. Annihilating 70 % of the tremors, this spoon is also suitable for people with hand tremors, a widespread neurological disease, not only among the elderly people.

Research Projects

The academic literature is also prolific on Smart-homes and AAL systems [48, 49]. Ambient intelligence can be used to monitor health status of the elderly with chronic diseases or, for assistance to people with physical or mental limitations, for rehabilitation, to improve the well-being of individuals, for their health education, etc. Table 2.2 shows an overview of the existing research platforms developed for AAL projects.

Historically, the original use-case for AAL is to assist elders in taking their medication [71, 72]. Since then, a large number of research have been conducted on smart reminders for elders' medication [73–76]. Alerts for the caregiver have also been integrated early on in AAL research [77, 78]. But, whereas alert are a critical short-term concern, other researches are rather interested in observing behaviour change for elderly people, over a long period of time [79–81].

In the past few years, we observe the integration of mobile application into e-Health and AAL systems [82].

Another promising extension of AmI is that of a generalisation from smart home towards smart cities, as reviewed in [83, 84].

[11] `http://www.liftware.com`

Research projects		Date	Specificity of the project
UbiSmart	[50]	2013	Smart home, web-based
UniversAAL	[51]	2012	AAL framework
Aging In Place	[52]	2011	Smart retirement house
SOPRANO	[53]	2009	Programmable smart home
OpenAAL	[54]	2009	AAL framewrork, supports SOPRANO
CASAS	[55]	2009	Adaptive smart home
Elite Care	[56]	2009	Sleep monitoring
DOMUS	[57]	2007	Focused on Alzheimer
CareLab	[58]	2007	Telecare system
CAALYX	[59]	2007	Wearable devices
WTH	[60]	2007	Electrocardiogram monitoring
HIS	[61]	2006	Evaluating health from patient's activty
ALARM-NET	[62, 63]	2006	Smart home architecture
Mav Home	[64]	2005	Agent-based system
Gator Tech	[65]	2005	Programmable and scalable smart home
ProSAFE	[66]	2005	Statistical behavior analysis
Ubiquitous Home	[67]	2004	Testbeds with complex sensors
Millenium Home	[68]	2004	Smart home
SELF	[69]	2000	Reporting physiological information
CareNet (MIDAS)	[70]	1998	Telecare system

Table 2.2: Existing research platforms in Ambient Assisted Living

Acceptability Of The System

According to Chernbumroong et al. [85], the main criteria for acceptability of AAL systems are *cost*, *accuracy*, respect of *privacy* and what they name *acceptance*. The concept of acceptance refers mainly to the low-visibility of the system. A system that would be too visible would by hardly accepted by the users. Therefore, AAL systems should try to remain low-profile, and not to stigmatize the user; wearable sensors should be used with care. To respect the privacy, one should avoid visual sensors such as cameras. In terms of cost, one should focus on low-cost sensors, with low battery consumption for minimal maintenance. Finally, the *accuracy* concern is tackled extensively in Chapter 7.

On the other hand, several researchers consider that the acceptability of a system is mostly driven by the questions of behavioral freedom, along with privacy and security [86, 87]. Ducatel argues that acceptability is driven by the utility of the system, its usability, and more generally the ability for ordinary people to benefit from it [88, 89]. Most researchers put an emphasis on privacy when questioned about acceptability of AAL [90–92], even suggesting a privacy-by-design architecture [93]. Legislations may also be an important safeguard for AAL systems [94, 95]. The security threats of AAL systems are discussed in [96–98], from the risk of mis-diagnosis to the sensitivity of health-related data gathered by such systems.

2.2 Technological Considerations

2.2.1 Architecture

By definition, ambient intelligence systems are pervasive, and composed of multiple objects. The architecture of such systems is usually 3-tier [99], as the user interfaces, the logical process and data acquisition consist of independent layers (Figure 2.2).

There are two main approaches in the development of such systems: centralized or distributed, which apply to different system layers. Figure 2.3 introduces the concept of 3-tier system with centralized or distributed approaches that apply to each architectural layer.

Following, I cover the literature related to the architecture of AAL systems, from the physical layers to the software implementation and the data engineering. This literature review has been useful for the development of the UbiSMART framework (Chapter 4)

2.2.2 Physical Layer

The physical layer is constituted by the hardware devices of the system. These are sensors integrated into the environment. Depending on the communication protocol, there are two types of sensor networks: wired and wireless (Figure 2.4). Wireless networks often complementary — rather than replacements — for wired networks. A wide hierarchy of wired networks is often used as the underlying network for wireless nodes (Figure 2.4a). This approach is used in most industrial applications, as it offers better performance and

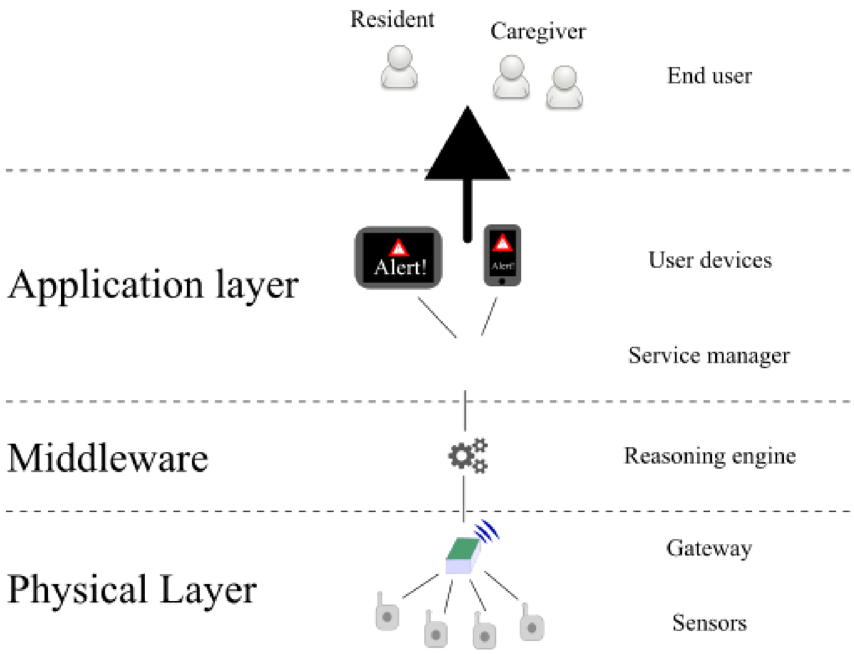


Figure 2.2: The 3-part architecture of a pervasive system

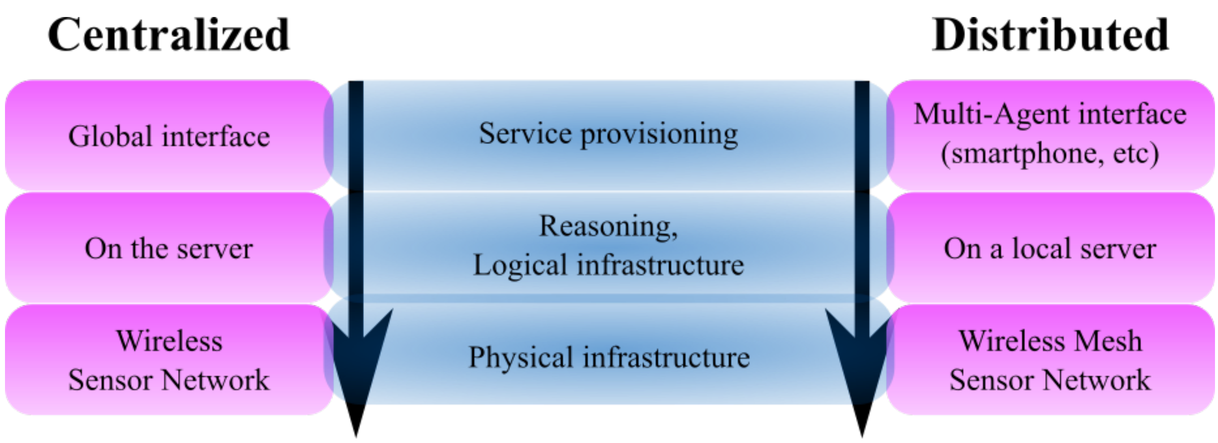
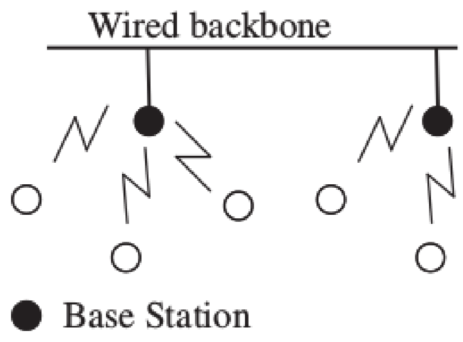
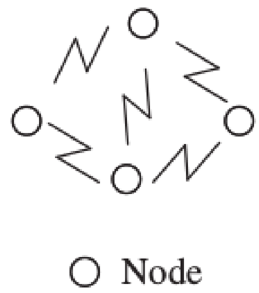


Figure 2.3: The 3-part architecture of a pervasive system: centralized or distributed

increased reliability. On the other side, in ad-hoc networks, all nodes maintain collaboration among them wirelessly [100] (Figure 2.4b).



(a) Wireless network with wired backbone



(b) Ad-hoc network

Figure 2.4: Sensor networks: wired and wireless

There are connections by Radio Frequency (RF), acoustic links, optical or infrared. Each connection has its advantages and limitations [101]. RF is the most widely used channel systems in sensor networks. Several aspects influence the radio power consumption, including the pattern of modulation type, data rate and transmit power.

Many prototypes and business solutions have adopted single transceiver radio for ad-hoc and low power consumption. Historically, one of the most famous solution is the X-10 RF protocol [12]. Other existing solutions are Z-Wave [13], EnOcean [14], Bluetooth Low-Energy [15], ANT+ [16], Libelium [17], and the very promising 6LoWPAN [18].

However, for several years already, the Zigbee protocol rises as a very popular solution for wireless communication, as it shows many advantages [38]:

– Low energy in communications;

– Longer battery life;

– Low-latency communication;

– 128bits security;

– Complete set of features;

– Large community.

[12] `http://www.davehouston.net/rf.htm`
[13] `http://www.z-wave.com/`
[14] `https://www.enocean.com/en/home/`
[15] `http://www.bluetooth.com/Pages/low-energy-tech-info.aspx`
[16] `http://www.thisisant.com/`
[17] `http://www.libelium.com/`
[18] `http://www.ti.com/lit/wp/swry013/swry013.pdf`

Distributed sensor networks do not need centralized access points, and are particularly suitable for dynamic and complex spaces. They are usually designed as Wireless Mesh Network (WMN) [102], based on mesh networking where each node must receive, send and relay data. This avoids having sensitive points, which in case of failure would isolate parts of the network. If a host is down, neighbors will go through another route. WMNs architectures can be classified into three categories:

– WMNs infrastructure where mesh routers form an infrastructure for clients;

– WMNs clients, where all clients are to perform the network functions;

– WMNs hybrid — a combination of the first two; therefore, clients can mesh functionality and access to the network.

One can find examples of using Wireless Sensor Networks (WSNs) in AmI environments in industry and in research. The ubiquitous computing project centered home "Oxygen" from the Massachusetts Institute of Technology [19], uses the N21 networks that connect mobile and stationary devices dynamically changing their configuration. The project APOGEE[20] from Siemens also automates the environment with a discreet and transparent system, yet reliable, thanks to the WMN technology. WSN use in healthcare applications has also been investigated [103, 104].

Sensors

Sensors for healthcare are mainly divided between body sensors [105–108] and ambient sensors [109].

Table 2.3 shows frequently used ambient sensors for AAL applications [110]. Several researches have included higher-level sensors to detect activities, through monitoring of TV usage [111], or smart-beds [112].

Sensor	Measurement	Data Format
PIR	Motion	Categorical
Active Infrared	Motion / Identification	Categorical
RFID	Object Information	Categorical
Pressure	Pressure on Mat, Chair, etc.	Numeric
Smart Tiles	Pressure on Floor	Numeric
Magnetic Switches	Door/Cabinet Opening/Closing	Categorical
Ultrasonic	Motion	Numeric
Camera	Activity	Image
Microphone	Activity	Sound

Table 2.3: Ambient Sensors Used in Smart Environments

[19] `http://oxygen.lcs.mit.edu/`

[20] `http://www.buildingtechnologies.siemens.com/bt/global/en/buildingautomation-hvac/building-automation/building-control/pages/building-control.aspx`

2.2.3 Software

In terms of software [113, 114], the major challenges are interoperability [115], modularity and dynamic provisioning of services, and remote communication. There seems to be an emphasis on Service-Oriented Architecture (SOA) for AAL solution [116–118], through the Open Service Gateway Initiative (OSGi) platform [119, 120]. RESTful implementations are also widely popular [121], and sometimes complementary to SOA architecture [122]. In [123], the authors choose to use the Devices Profile for Web Services (DPWS) protocol to integrate devices into their system.

Kim et al. propose an extensible design to integrate seamlessly and safely multiple devices and protocols into a AAL system [124].

2.3 Activity Recognition

Activity Recognition is central in Ambient Assisted Living. Every action taken by the system is determined by knowledge of the context, and more specifically by the activity being recognized. Thus, in AAL, each object is considered not as an individual device but as part of the global environment, contributing to a part of the overall problematic.

The Activity Recognition process should be considered as an application of artificial intelligence. Using a simplistic paradigm for artificial intelligence, most objects can be described by a set of properties, methods and events. Using a “smart water tank” object as an example, one could define a property “water level”, as well as methods to “open bolt” and “close the cylinder head” to control the fluid supply. This intelligent tank can also generate events such as “virtually empty”, “almost full” and “water level exceeded”.

Another famous pattern of artificial intelligence is the Performance Environment Actuators Sensors (PEAS) [125]. According to the PEAS pattern, every application can be modelled as four sets of parameters:

Performance The system needs to be given an indicator of performance, so that it will approach optimal reaction towards a problematic. As for AAL, the performance can either be the accuracy of Activity Recognition, or — more ambitiously —, the impact of the system on the end-users’ life.

Environment This parameter refers to the knowledge of the environment at a specific instant. In AAL, this is usually referred to as context-awareness.

Sensors Sensors are the inputs used by the system to update its knowledge of the environment. I have covered the use of sensors in AAL in Section 2.2.2.

Actuators Actuators are the media available for the system to interact with its environment. In the case of AAL applications, it may be a phone alert, a message on a smart TV, etc.

Artificial intelligence is mostly applied towards task planning. Planning can be used in different ways: to plan daily activities in a flexible way and help users to perform their daily activities, to automate the daily routine, etc. AmI applications bring new challenges to traditional planning techniques. Planning must be adapted to a dynamic environment with incomplete information, where the result of actions and their duration is not fixed. Many planning techniques have been proposed in AmI. Arguably one of the most appropriate is the Distributed Hierarchical Task Network (D-HTN). It uses a centralized approach for managing data from devices distributed [126]. This technique was applied to intelligent houses for diabetic patients, where all devices communicate with each other to plan actions such as temperature control, insulin injection etc. [127]

Nevertheless, the real challenge of AAL is less of planning activities than it is of Activity Recognition. One can refer to *task projection*, that is the inverse of task planning: whereas task planning tries to generate a sequence of tasks, projection tries to deduce this sequence from observation [128]. In the case of AAL, the most widely accepted units of context are activities and actions — an activity being composed of several actions.

2.3.1 Activities, Context and Situation

Before starting to tackle the topic of Activity Recognition itself, I find it necessary to define clearly the concepts of *data*, *action*, *activity*, *context* and *situations*.

Situation

Barwise and Perry [129] propose the following definition of a situation:

> The world consists not just of objects, or of objects, properties and relations, but of objects having properties and standing in relations to one another. And there are parts of the world, clearly recognised (although not precisely individuated) in common sense and human language. These parts of the world are called situations. Events and episodes are situations in time, scenes are visually perceived situations, changes are sequences of situations, and facts are situations enriched (or polluted) by language.

Context

While interesting, the definition *situation* above remains rather philosophical, and cannot be easily formalized as a model. Therefore, most researchers prefer to use on the notion of *context*, which is complementary to *situation* (I follow the same approach in this thesis document).

The most widely accepted definition of context is the one of Dey et al. [27]:

> Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

One may notice that this definition of *context* is largely equivalent to the concept of *environment* in the PEAS paradigm.

It is important to notice that, practically speaking, contexts are not necessarily unique nor absolute. One may actually isolate primary context — that is derived directly from sensors — from secondary context — that is inferred from several data stream (including lower-level context). Contexts are even dependent from one another, as they can be bound together by several relations [130, 131]:

Generalisation A context can be regarded as more general than another context, if the occurrence of the latter implies that of the former. The opposite relation of *generalisation* is *specification*. For example, a "watching TV" context is considered more specific than an "entertainment" context, because the conditions inherent in the former context are a subset of conditions in the latter context [132].

Composition A context can be decomposed into a set of smaller contexts, which is a typical composition relation between contexts. For example, a "cooking" context is composed of a "using stove" context and a "retrieving ingredients" context. McCowan et al. propose a two-layered framework of contexts: a group context (e.g., "discussion" or "presentation") is defined as a composition of contexts of individual users (e.g., "writing" or "speaking") [133].

Dependence A context depends on another context if the occurrence of the former context is determined by the occurrence of the latter context. Dependence can be long- or short-range, as proposed by [134]. Sometimes long-range dependence can be useful in inferring high-level contexts. For example, a context "going to work" may be better in inferring a context "going home from work" than other short-range dependent contexts.

Contradiction Two contexts can be regarded as mutually exclusive from each other if they cannot co-occur at the same time in the same place on the same subject; for example, a user cannot be in a cooking context and a sleeping context at the same time.

Temporal Sequence A context may occur before, or after another context, or interleave with another context; for example, "taking pill" should be performed after "having dinner" [135].

For a context model to render the full spectrum of real-world context, Bettini et al. suggest that it implements the following requirements [136]:

- Heterogeneity and mobility;
- Relationships and dependencies;
- Timeliness;
- Imperfection;
- Accepting Reasoning;
- Usability of modelling formalisms;
- Efficient context provisioning.

Activity and Actions

In AAL, activities are the most critical elements of context. Whereas a context may include multiple types of informations — such as weather, time of the day, or even spatial information such as the topology of a room —, an AAL system is mostly interested by the *activities* of the end-user. More precisely, we often refer to ADLs, simply defined as follows [21]:

> Routine activities that people tend do everyday without needing assistance. There are six basic ADLs: eating, bathing, dressing, toileting, transferring (walking) and continence.

On the opposite, *abnormal activities* are unexpected activities, and are described more extensively in the Section 2.3.3.

An activity can be modelled as a sequence of atomic actions. However, the model has to be flexible enough to accept that one activity can be performed in several ways, and that users sometimes even do several interwoven activities at once.

Activities can be defined on various granularities, and we can also mention higher-level activities, that would be composed of several lower-level activities.

Sensor Data

Sensor data are defined as "raw (or minimally processed) data from physical and virtual sensors".

The DIKW Model

Now that defined every important concepts are defined — from a *situation* to *data* — it may be interesting to introduce the Data-Information-Knowledge-Wisdom (DIKW) model (Figure 2.5). This model simply states a hierarchy between *data*, *information*, *knowledge*, and *wisdom*; each of them defined on top of the former one.

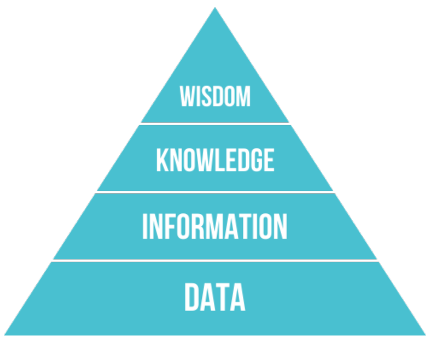


Figure 2.5: The DIKW pyramid

[21] http://www.investopedia.com/terms/a/adl.asp

Activity Recognition vs. Context-Awareness

In the literature, context-awareness and activity recognition sometimes are used as equivalents. However, given the definitions above, context-awareness is more general than activity recognition. In this thesis, although I need a certain level of context-awareness to recognize activities of the residents, my primary target remains activity recognition. Therefore, in the remainder of this document, I choose to use the practical term of *Activity Recognition*, rather than the concept of *Context-Awareness*.

2.3.2 Methods Of Activity recognition

Different approaches exist for Activity Recognition, depending on technologies used for sensors, on the complexity of the activities of the person, and on the final objectives [137]. Two main approaches exist to design the system for recognition of activities and decision support: Knowledge-based approach, and data-driven approaches (often assimilated to machine-learning) [28]. In this thesis work, I do not advocate for a method in particular (although I make use of knowledge-driven reasoning). Still a review of these various methods is important in order to understand how this thesis' contributions may — or may not — be applied to a particular approach for Activity Recognition.

Knowledge-Based Reasoning

Logical Programming

In the early stage of activity recognition, there was a low number of sensors, and observed situations were rather simple, so they could be easy to model with formal logic, in order to have an expressive and sound reasoning [138].

The work of Loke et al. [139–142] aims to reason at a high-level of abstraction, over a declarative approach. For example in [141]:

```
if in_meeting_now(E) then
    with_someone_now(E) ,
    has_entry_for_meeting_in_diary(E) .
if with_someone_now(E) then
    location*(E, L) , people_in_room*(L, N) , N > 1.
if has_entry_for_meeting_in_diary(E) then
    current_time*(T1) ,
    diary*(E, 'meeting', entry(StartTime, Duration)) ,
    within_interval(T1, StartTime, Duration) .
```

Similarly, in [143], the authors defines roles and relations of an AAL system, in order to characterize activities and situations Kalyan et al. [144] define a multi-level situation theory, using the concept of *infons*, *micro situation* and *situation*.

For a better abstraction, existing logic theories from Artificial Intelligence (AI) research have also been used, such as situation calculus [145, 146], that was applied to AAL in [147, 148].

Spatio-Temporal Logic

Spatial and temporal logics are already well developed in the field of AI. Allen's Interval Logic [149] describes relations between multiple qualitative temporal event (Figure 2.6). RCC8 [150] is often used alongside with Allen's Interval Logic, and describes qualitative spatial positioning (Figure 2.7). Their usage are relevant in the context of AAL, as spatial and temporal information are essential characteristics of an activity.

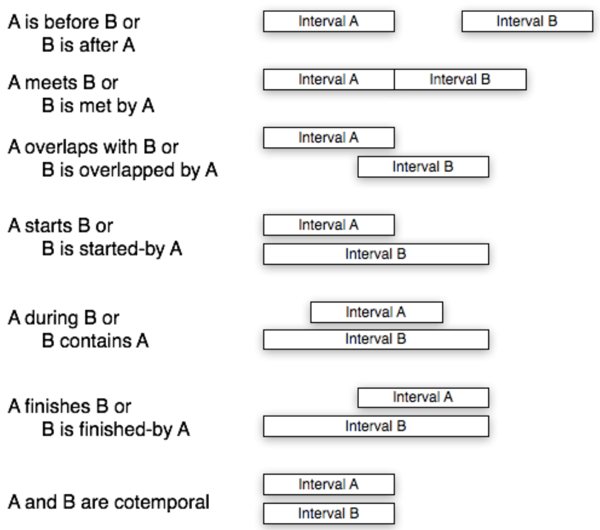


Figure 2.6: Qualitative Temporal Logic with Allen's Interval Logic

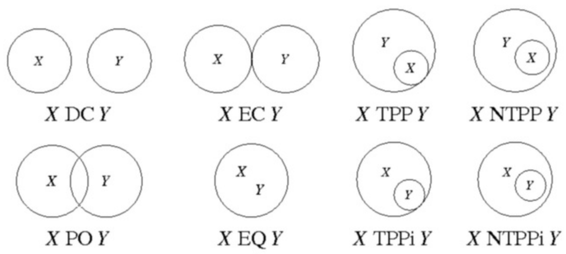


Figure 2.7: Qualitative Spatial Logic with RCC8

Gottfried et al. [151] use these qualitative spatio-temporal techniques to perform activity recognition within smart-homes.

Ontologies

Although basic logical approaches and spatio-temporal logics introduce interesting theories, they are limited in their expressiveness, and often lead to unintuitive solutions. Therefore, most researches in knowledge-based reasoning are focused into ontological reasoning.

The central theory of the *ontological reasoning* approach claims that knowledge has two main components: an explicit part with the definition of the knowledge (ontology base) and a tacit part, which can be extracted from inferences engines [152]. An ontology contains definitions and instanciations of *Things*, as well as associations between these things, and *IF-THEN* rules. At a more fundamental level, even rules, association and abstract concepts of definition are *Things*. The inference engine then combines rules with data from actual sensors to emerge a new tacit knowledge. The choice of a suitable combination of a reasoning engine, a communication interface and the ontology expressiveness used is a complex task, which investment in time and energy is often underestimated [153].

Ontologies provide methods to keep the facts of the real world into logical form. The definition of ontology as a computer term, is "an explicit specification of a conceptualization" [154]. There are two types of ontologies: high-level, that describe the general knowledge, and specific, that describe the concepts specific to the area of interest [155]. Using Web Ontology Language (OWL) [33], one can map domain specific ontologies onto the high-level ontology of general knowledge. This helps to focus the efforts on modelling the environment rather than on defining a model of the global universe.

There are several ontologies for intelligent environments, each with a slightly different focus [156–166]. Some, like DogOnt, specify interoperability between home automation systems [167], and consequently its features and capabilities, others, like COSE [155] focus primarily on modeling objects and sensors in intelligent environments. In the case of complex environments, it is also possible to use multiple concepts from a variety of ontologies.

Several attempts have been made to tackle the complexity of real-life situations [168–171], by modelling objects interacting with the user [172], the sensors [173], by defining a user profile [174], or by providing strong spatial knowledge of the environment [175]. Yau et al. includes a rich semantic of situations in their ontology [176], where situations can be hierarchical, connected to each other, subsequent, etc.

In term of implementation, ontologies are defined using the Semantic Web technologies, and the OWL language [177, 178]. To permit an automatic reasoning over the ontology, OWL has to be constrained to a Description Logic (DL), as is OWL-DL [179]. To overcome some limitations of OWL-DL, the SWRL language has been developed [180], as applied in [181]. Additionally, the N3 formalisms provides a comprehensive syntax and powerful capabilities on top of classic Resource Description Framework (RDF) [32].

The reasoning engine ensures the propagation of knowledge in the system, so it is important to choose one that meets the platform requirements deployed: the soundness of knowledge, a reasonable complexity in terms of time and resources, extensibility for large-scale deployment, reliability rules. A comparison of the major reasoning engines is given by Tiberghien et al. in [182] (Table 2.4). Based on this benchmark, we can observe the superiority of Euler reasoner [34] over its competitors.

	Jena	**Pellet**	**RacerPro**	**Euler**
OWL-DL entailment	Incomplete	Full	Full	Full
Rule format	Own, basic & built-ins	SWRL	Own, Powerful	N3 & Built-ins
Retractability	Yes	Can Emulate Stateless	Yes	Stateless
Ease of use	Average	Easy	Complex	Easy
Response time for 100 triples	783ms	442ms	503ms	4ms
Response time for 1,000 triples	22,330ms	38,836ms	44,166ms	40ms
Response time for 10,000 triples	Out of Memory	Out of Memory	Out of Memory	436ms
Scalability	Very limited	Average	Limited	Good
Size (download)	22.3Mb	24.3Mb	60.3Mb	12.9Mb
Licensing	Freeware, Open Source	Freeware, Open Source	Shareware, Closed Source	Freeware, Open Source

Table 2.4: A comparative table of semantic reasoners (2012)

Dealing With Uncertainty

The major weakness of knowledge-based approaches for AAL is that they don't natively reason under uncertainty. Adding uncertainty to an ontological model is very uneasy, given the constraints that were given to its DL to retain decidability. Several research works have been done to extend ontological languages so that they would support fuzziness and uncertainty, while remaining decidable [183, 184].

On the other hand, the work of Lei et al. [185] declares uncertainty metrics into an ontology, namely *accuracy*, *incompleteness*, *timeliness* and *reliability*. Similarly, in [186] the authors introduce the notion of *time decay* to reduce the confidence of sensor data as they become old (and most likely outdated).

As a different approach [187] uses ontological reasoning in a unusual way, not to perform activity recognition, but to validate activities inferred by statistical method.

But the two most promising approach to perform reasoning with uncertainty in an ontological model are Fuzzy Logic and Dempster-Shafer Theory (DST).

DST [188, 189] is used to perform data fusion. This way, redundant information between several sensors data, each with its own confidence level, can be merged to emerge a higher-level information, with a reasonable confidence given the data sources' confidence. It has been used by [190–193] in the context of AAL. Aloulou et al. [194] propose a model where the uncertainty is attached to the relation between two entities, rather than being attached to the entity itself. Through this design, they are able to propagate the uncertainty caused by sensor defects into high-level activities, with the use of DST.

Fuzzy Logic [195] attaches imprecise and intuitive notions of confidence, such as *trustworthy*, *unlikely* to events, and provides tool to reason over them. It has been used by [130, 196, 197] to integrate uncertainty to activity recognition.

Other approaches for reasoning under uncertainty are probabilistic logic [198], sensor data fusion with Bayesian Network (BN) [199], or diagnosing the sources of fault in the sensors with BN [200], although these two last methods are hybrids between knowledge-driven and data-driven approaches. Similarly, in [201], the authors propose an architecture that would integrate different reasoning mechanisms, from Fuzzy Sets and DST on the sensors level, to higher level BN.

Data-Driven Reasoning

The data-driven approach mostly uses machine-learning algorithms to exploit the data.

A comparison of different knowledge extraction techniques related to the IoT was given by Chun-Wei Tsa et al. [202]. To determine which technique is applicable to a given problem, they propose the matrix in Figure 2.8. There are two types of methods for activity recognition in this case: the identification of pre-defined activities, based on the techniques of *supervised learning*, and the discovery of unknown activities, based on the techniques of *unsupervised learning* [55, 203]. Depending on whether the sequential aspect of the observed events, we can either apply *Association Rules* if we need to find events from entries without special order, or *Sequential patterns* if we need to find events from entries with a particular order.

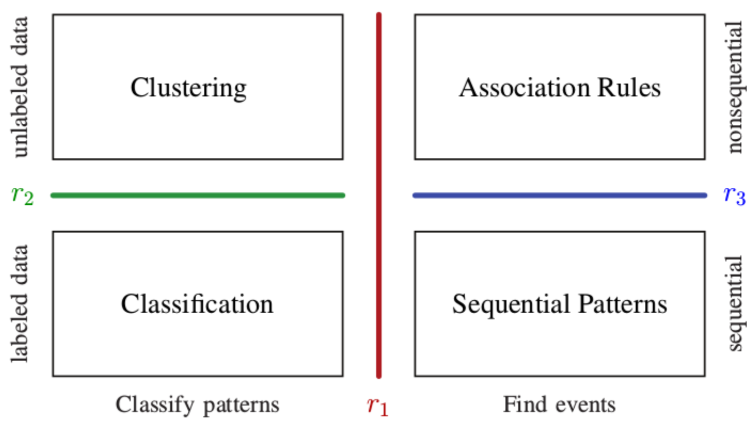


Figure 2.8: The matrix of knowledge extraction methods.

Table 2.5 (partially inspired by the work of Ye et al. [28]) summarizes the literature of data-driven activity recognition in AAL, based on the categories described is Figure 2.8. The most significant methods of Table 2.5 are described following.

These methods vary depending on the environment, the amount of data and objectives of the modeled system. Generally supervised learning techniques are divided into template matching, generative models and discriminative models. The methods called template matching use nearest-neighbor classifiers, and a multi-instance learning with labeled bags [224]. The generative models are probabilistic models for stochastic data,

Learning Technics	Supervised	Unsupervised	Static	Sequential
Naive Bayes [204–208]	✓		✓	
Bayesian Networks [132, 184, 200, 209–214]	✓		✓	
Decision Trees [215–218]	✓		✓	
Support Vector Machines [204, 219–221]	✓		✓	
Emerging patterns [222]	✓		✓	
Template Matching [223–225]	✓		✓	
Case-based reasoning [226]	✓		✓	
Hidden Markov Models [227–240]	✓			✓
Conditional Random Fields [241–248]	✓			✓
Context-free Grammar [249, 250]	✓			✓
Intertransaction association rule mining [251]	✓			✓
Markov Decision Process [252]	✓			✓
Neural Networks [253]		✓	✓	
Web mining [254, 255]		✓	✓	
Jeffrey Divergence [204]		✓	✓	
Activity Stream Mining [256]		✓	✓	
Semi-Supervised Learing [257]		✓	✓	
Active Learning [258]		✓	✓	
Low-Dimensional Eigen-Spaces [259]		✓	✓	
Mixture models [260]		✓	✓	
Patterns Regex [260]		✓	✓	
Constraint Based Mining [261]		✓	✓	
Frequent Periodic Pattern Mining [55, 262]		✓	✓	
Gais Genetic Algorithm [203]		✓	✓	
Activity Episode Discovery [262]		✓		✓
Suffix trees [263]		✓		✓
Sequencial Activity Mining [137, 256, 264, 265]		✓		✓

Table 2.5: Supervised vs. Unsupervised and Static- vs. Sequential-Based Learning

usually given some hidden settings. For instance, Hidden Markov Models (HMM) and Dynamic Bayesian Network (DBN) can extract sequences of activities even in noisy environments [209]. Discriminative models — also known as conditional models — include Conditional Random Field (CRF) classifiers and Support Vector Machine (SVM). They are effective in modeling groups of separate activities, and help to overcome simplifications with representations of contextual dependencies for more precise recognition [224, 266].

Such supervised learning techniques learn the models and parameters from training dataset, that usually requires a human to label a situation to sensor data that are observed during the occurrence of this situation. In the case of AAL, the labeling process is often impossible, or limited to lab-deployment, whose data are sometimes unrealistic. Therefore, supervised learning techniques may have limitations in real-life deployment where scalability, applicability, privacy, and adaptability are highly concerned [267].

To tackle this issue, researchers have employed unsupervised learning approaches. Among them, suffix-tree and Jeffrey divergence can extract features from sensor observations, which are distinguishable from one situation to another. The learning results can be directly used to infer human activities [255] or as a source for other supervised learning techniques [267] Recent techniques of unsupervised learning extend conventional approaches (titled-time window based on constraints, mining frequent periodic patterns using regular expressions, etc.) for the search for more complex patterns — discontinuous and disordered — with automated labeling. Using this approach, manual annotation of activity data is no longer needed [137, 256].

One of the first use case of data-driven reasoning for activity recognition was the implementation Adaptive Home [268], which used the neural network and machine learning to control lighting and temperature of the water in order to reduce operating costs. One approach taken by the project iDorm [269], is to use an expert system with fuzzy logic to learn the rules that reproduce the interactions between inhabitants and home automation devices. These projects were the basis for more complex environmental modeling projects with multi-modular architecture.

A learning approach based on a Hierarchical HMM [234] has helped to create an environment that is able to learn the behavior of people and plan their activities. This system supports services with a multi-modular architecture using the Semantic Web for much of the intelligence layer: using an ontology module, the reasoning identifies prominent characteristics in the context of the patient, and then tries to take the best action according the different motives of the patient's behavior.

Another promising solution for optimization and knowledge extraction in the IoT is the use of meta-heuristics algorithms [270]. They model the problem as a search for the best path in a graph representing the different states of the problem.

2.3.3 Abnormal Activity Detection

While it is highly valuable to recognize Activities of daily Living, healthcare applications are particularly interested in abnormal events, indicating a crisis or a change for the patient. Yet, the concept of anomaly remains rather unclear.

A general classification of anomalies is given by Chandola et al. [271], where the authors mention the following anomalies:

Point Anomaly These are punctual values that are never abnormal with respect to the rest of the data. That could be an accelerometer suddenly measuring speed of 30km/h when it usually measures speeds of 2-5km/h.

Contextual Anomaly These are data instances that are abnormal in a certain context, but not otherwise. That would be the case of a stove being turned on in the middle of the night.

Collective Anomaly Collective anomaly are a collection of data instances that together become abnormal. A good example would be a motion sensor in the bathroom being activated for five hours.

On a higher-level, new behaviors are often caused by an unusual context: a window opened during a storm, not opening the refrigerator for a day, tap opening night and not after closing, etc. There are many possible interpretations for abnormal activities. Therefore, I appreciate the following classification of anomalies [272], which is more human-centered:

1. Statistical anomaly. Users are given a normal distribution of their activities, and the behaviors located away from this curve are considered abnormal;

2. The deviation of socially accepted standards;

3. The personal development theory: if a normal evolution can be defined, therefore the failure to evolve the predicted way is considered abnormal;

4. Subjective anomaly: when the person feels a personal anomaly, including anxiety, depression etc;

5. Biological deviations: diseases or organ injury.

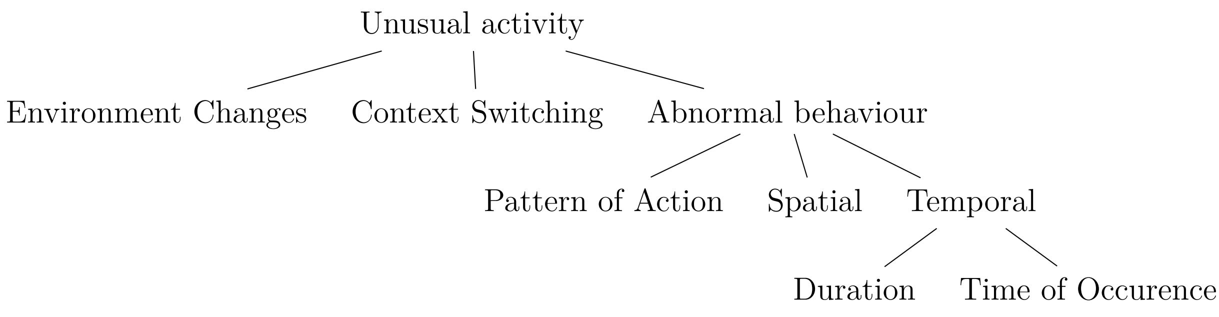


Figure 2.9: Use cases for abnormal human behaviours

Tran et al. [273] define a variety of use-cases to ensure the identification and isolation of abnormal behavior (Figure 2.9).

Several attempts have been made for unsupervised abnormality detection, using statistical methods [274] or data-mining algorithms [275, 276], usually focusing on temporal or sequential anomalies. However, a vast majority of the literature seems to focus on a supervised method, where abnormal activities are explicitly defined. This is the case of fall detection [277–282], or when detecting elderly people wandering out [283–286].

For these explicit definitions of anomalies, the problem is no more than a problem of activity recognition, as described in Section 2.3.2. However, as opposed to routine activities like *sleeping* or *eating*, anomalies are unusual. Therefore, they will generate smaller training sets, and are subject to statistical bias, and Data-Driven methods will show difficulties to train properly and recognize these activities correctly.

2.4 Challenges Of Activity Recognition

2.4.1 Complex Activities

Arguably the largest challenge of AAL is the complexity of the context being observed. The physical world acts as a dynamic environment with incomplete information, and emerging data often generates complex patterns. More specifically, I have isolated four challenges for Context-Awareness in real conditions. I do not claim to be exhaustive through these for challenges, and my purpose is to illustrate the overall complexity of Context-Awareness, not to provide a thorough listing of its challenges.

Real-Time Processing

Most machine-learning algorithms are designed to be run offline, whereas AAL systems require activity recognition to be processed in real-time [256]. However, the recent development of Big Data and its application for Web Mining have greatly improved the ability of Data Mining algorithms to be performed in real-time.

Transferring Knowledge Between Two Environments

Designing and training an AAL system requires a large amount of data. However, every environments are different. The topology of sensors deployed will vary from one house to another, some residents will have strong routines where others don't, etc. Taking this variety of environments into account, one needs to be able to transfer knowledge from one environment to a new one, and keep the training payload as little as possible [287–290].

Interwoven And Concurrent Activities

There are several ways to make tea. A well trained system should be able to recognize the most usuals of them. But sequences of actions can often be switched. For instance, one can take the glass before starting to boil the water, or after. More generally, two activities can be performed concurrently. One can prepare tea while cooking bread toast. The two sequences of action will be interwoven, therefore disturbing most algorithms for Activity Recognition [236, 291, 292].

We have seen in Section 2.3 that rich relations can exist between contexts, such as *Generalisation* or *Dependence*. A first step towards recognition of such activities would be to define a rich semantic model able to integrate interwoven activities.

Multiuser Situations

The question of multiuser situations [293, 294] is complementary to the one of interwoven activities, as it also involves parallel sequences of actions. This problem can be easily solved with the use of cameras, or by Radio Frequency Identification (RFID) wearable sensors, but these sensors are rarely accepted by the residents, as the action of constantly checking-in is their own house can hardly be accepted. In [295], we suggest a natural reaction towards multiuser situation: rather than recognizing activities with multiple users, we simply detect whether there are multiple users in the residence or not. If there are, we consider that the resident is accompanied with a caregiver, and therefore that he is safe (Chapter 6).

2.4.2 Switching Towards Real Deployments

Since the beginning of AAL, real applications have been considered, particularly into nursing homes [296]. Similarly, Panek et al. [297] share their experiences of a user-centric deployment inside living labs. [298, 299] also provide two insightful experience feedback about development of a smart-lab. However, these are living labs, not real nursing homes. Chung et al. [237] gives an interesting application of activity recognition inside a nursing home.

According to the survey conducted by Memon et al. in 2014 [300], we currently observe a shift in AAL research, from laboratory experiments towards real deployments. Rather than focusing on raw theoretical concerns, they suggest for researchers to put their focus on:

- Design methodologies;
- User experience;
- Usability;
- Security;
- Dependability;
- Accuracy.

However, still based on Memon et al. survey [300], there exists a large number of implementations of AAL, but only a handful of them have real world usage. The authors therefore argue that very few teams have conducted real-world deployments and gathered significant data.

In my humble opinion — based on our own experience with real deployments, and on the insights we gathered from [1] — large efforts in AAL are currently towards the technological aspects, but less efforts are directed towards equally important real matters: usefulness, usability, security, acceptability, maintainability, reliability, quality of the user experience.

2.4.3 Uncertainty Of The Reasoning Process

For AAL, uncertainty is natively handled by many data-driven algorithms like HMM (Section 2.3.2), and we have covered uncertainty management in knowledge-based reasoning in Section 2.3.2.

The quality of data, information and contextual knowledge is crucial in AAL systems according to McNaull et al. [301]. They propose four layers of the AAL system architecture including base, data, information and context layers to investigate the quality attributes of sensors, ambient data, and communication interfaces. Aloulou et al. [194] tackles uncertainty from a sensor level, by giving a confidence to each sensors. Similarly, Henricksen et al. [302] states that sensor data can be *out of date*, *incomplete*, *imprecise* or *contradictory*.

Another critical process to deal with uncertainty in Activity Recognition, and a key step towards wide acceptance of AAL, is the validation of reasoning engines, by measuring their accuracy in real situations. The legacy approach for assessing the ADLs of elderly people, whether by direct observation [303] or by questionnaires [304–306], are instructive, but they lack of practical applications towards activity recognition in the AAL. Kleinberger et al. [307] performed a thorough validation of their system, measuring its accuracy with the use of the well-established Goal-Question-Metric (GQM) approach [308]. They observe an accuracy of 92% on average for simple ADLs such as “going to toilet”. However, although their methodology is interesting, we can note that they only validate three activities, taking the hypothesis that laboratory setup is representative of real environment. This hypothesis is reasonable for these three simple ADLs, but I am unsure that it can be extended to more complex and long-term activities. Since 2009, several other researchers have used similar methods to validate their reasoning engines over laboratory experiments [85], with more than 90% of correct inferences.

2.5 Discussion

AAL is a quickly expanding topic. There exist multiple promising applications, emerging from the industry as well as from academic researches. However, only little focus is given on several important aspects of AAL: *usefulness*, *usability*, *acceptability*, *reliabilty*, etc. Although we observe a recent trend towards real-world deployments, only a few research projects have actually deployed their systems in real environments, and gathered sufficient material from these deployments.

In this thesis, I introduce our platform for AAL — UbiSMART — that was developed to address the needs observed from the real-world deployment we have performed.

Activity Recognition is a critical component of AAL. A majority of researches use data-driven methods such as HMM or data-mining algorithms, but knowledge-based reasoning are not to be ignored, as they have more flexibility to cope with complex models, such as models of smart homes. A key step towards wide acceptance of AAL solutions is the evaluation and validation of reasoning engines [309]. Several teams have validated their reasoning engines over test sets and laboratory experiments, measuring the accuracy of their reasoning engine in recognizing activities. These accuracy results are solely used to indicate how well does a reasoning engine perform. To the best of my knowledge, they have not yet been used in a systematic process to improve the reasoning.

In this thesis I introduce a method for improving the reasoning. This method is reasoner-agnostic; it can be applied with any of approaches for Activity Recognition (e.g., knowledge-driven approach, DBN). The method improves the decision-making process on the output of the reasoning (i.e., the set of activities possibly being performed by the end-user), through Accuracy, Precision and Score of the reasoning engine.

This thesis work also relates to a higher-level problematic of Activity Recognition, that is facilitating the development of a reasoning engine, in order for the researchers to perform fast-prototyping for Activity Recognition.

Seek first to understand,
then to be understood.

Steven R. Covey

3 User Needs Investigation

3.1 Survey – The lifestyle of the elders

In order to understand the lifestyle and needs of elderly people within their home, we have created a questionnaire, in collaboration with a French national association of retired people[1] [1]. In this questionnaire, the same questions have been asked to both the elderly people and their family caregivers, so that we can observe the bias in their perception. We have interrogated 123 couples elderly people / caregivers (i.e., 246 questionnaires), for a total of 35,178 questions answered. From this survey, we aim to determine the most critical needs of elderly people in their daily lives.

The content of these questionnaires is provided in Appendix A.

3.1.1 Content Of The Questionnaires

The two questionnaires (for the elderly people and for caregiver) are 14 pages each, and are divided into five sections:

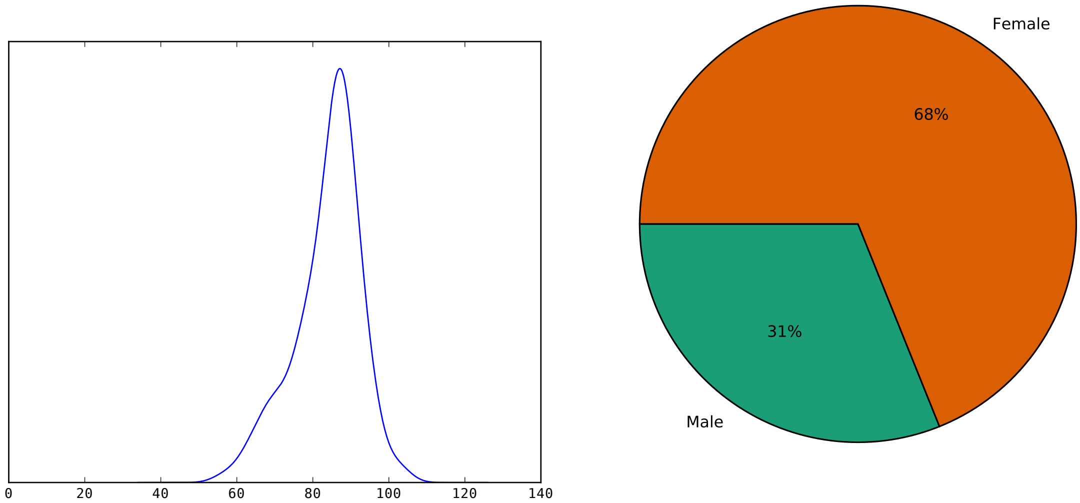
- The situation of elderly people and caregivers;
- The daily activities of the elderly people in his house;
- The outdoor activities of the elderly people;
- The social and financial helps received by the elderly people;
- The expectations of the elderly people towards our services.

[1] Association Nationale des retraités de La Poste et de France Télécom (ANR)

3.2 Population Studied

3.2.1 Elderly People

As shown in Figures 3.1a and 3.1b, most of the elderly people surveyed are women, and are aged around 85 years-old.



(a) Age of the elders **(b)** Sex of the elders

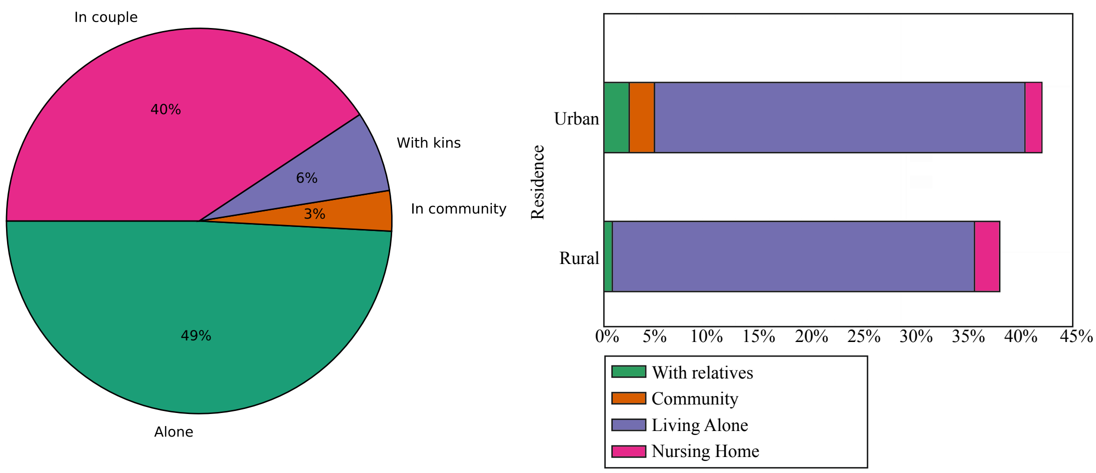
Figure 3.1: Elders studied in the questionnaire

Half of the elderly people surveyed live alone in their residence (Figure 3.2a). Another 46% are living with their companion or with their kins, and only 3% live in a community (e.g., nursing home). The apparent low number of elderly people who live in a nursing home is due to an expected bias, as this population is not the primary target of the survey.

Figure 3.2b shows that there are almost as many elderly people living in countryside as those living in city environment. As compared to the French population, where 80% of people are living in city environment, we can consider that a larger number of elderly people are living in countryside. When considering an AAL system, one must integrate this information. Countryside environments often have less access to the Internet, and less medical facilities.

3.2.2 Caregivers

Based on Figures 3.3a and 3.3b, most of the caregivers are women and are aged between 50 and 80 years old, except for the professional caregivers (11% of them, based on Figure 3.3c). Additionally to the professional caregiver, 35% of the caregivers are the elders' children, and 25% are the elders' companions. The remaining 25% are either friends, other members of the family, or community members. I therefore distinguish 3 categories of caregivers:



(a) Persons living with the elders **(b)** Residence of the elders

Figure 3.2: Daily environment of the elders

- The professional caregivers, aged around 40 years old;
- The elders' children, aged around 60 years old, who are women for most of them;
- The friends and relatives, who are generally aged around 80 years old.

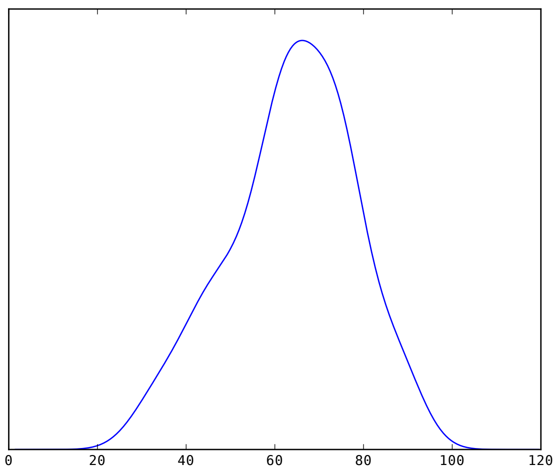
Thus, as confirmed in Figure 3.3d, most of the caregivers are retired persons themselves. This is an important indicator on their availability for the elderly people.

Help provided

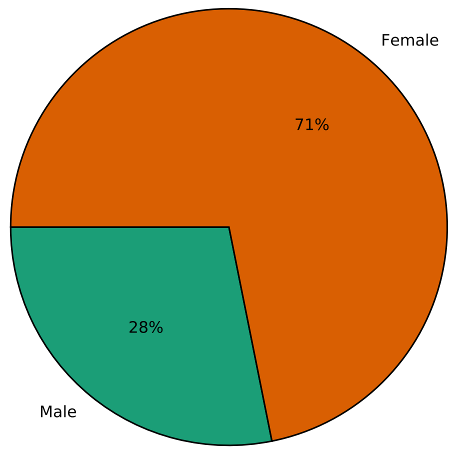
Figure 3.4a confirms that caregivers provide numerous helps to the elderly people. A large part of caregivers even respond that they provide every kinds of aids, as required. Apart from this general support, the main role of caregivers is moral. Second comes social assistance (bringing elderly people to the doctor, transportation). Then comes health-care assistance (walking, hygiene, waking up the resident, taking medications, meal preparation), logistics (organization of the schedule, help with paperwork) and material (gardening, food shopping...).

The number of hours spent with the elderly people, shown in Figure 3.4b, is divided into two poles:

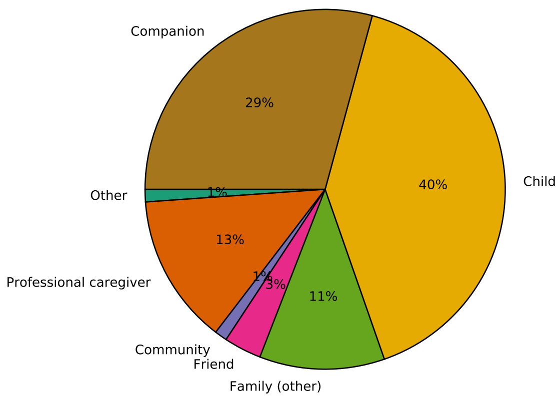
- Caregivers living at home, which accompany the elderly person throughout the week, day and night;
- Distant caregivers, professionals or not, who accompany the elderly person during 0 to 30 hours each week.



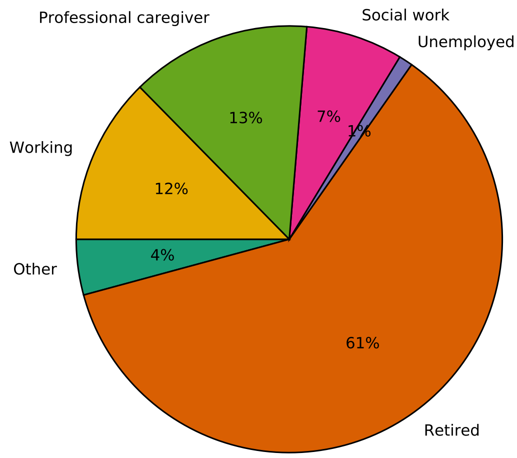
(a) Age of the caregivers



(b) Sex of the caregivers

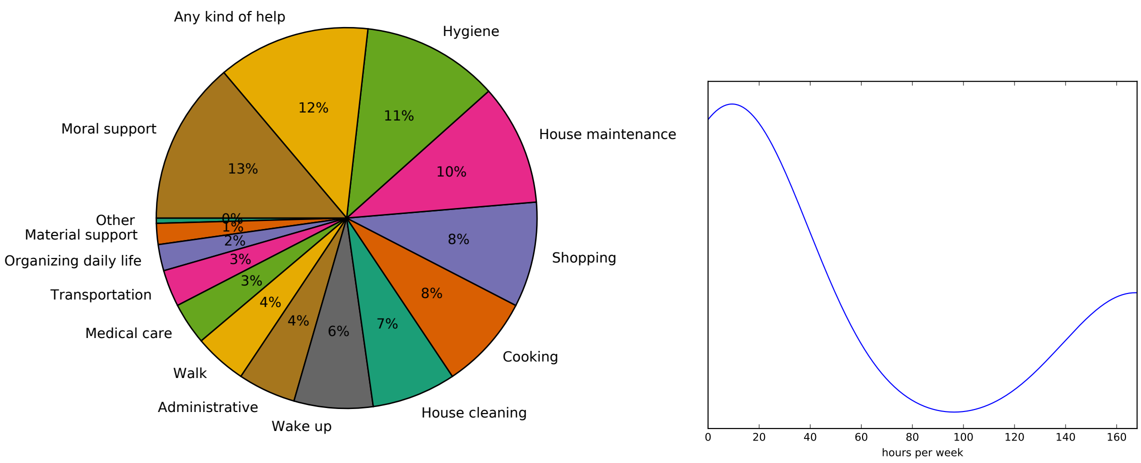


(c) Relation of the caregiver with the elder



(d) Profession of the caregivers

Figure 3.3: Situation of the caregivers



(a) Help given to the elderly people, by the caregivers **(b)** Weekly hours spent by caregivers supporting the elderly people

Figure 3.4: The roles of caregivers

Distance between the elder and the caregiver

According to Figure 3.5, 35% of caregivers share their home with the elderly person they accompany. In total, more than 80% of caregivers live within 10 km of the elderly person. However, for the 13% of caregivers who live more than 50 kilometers of the elderly person, they can not grant him frequent visits, nor respond quickly in case of emergency.

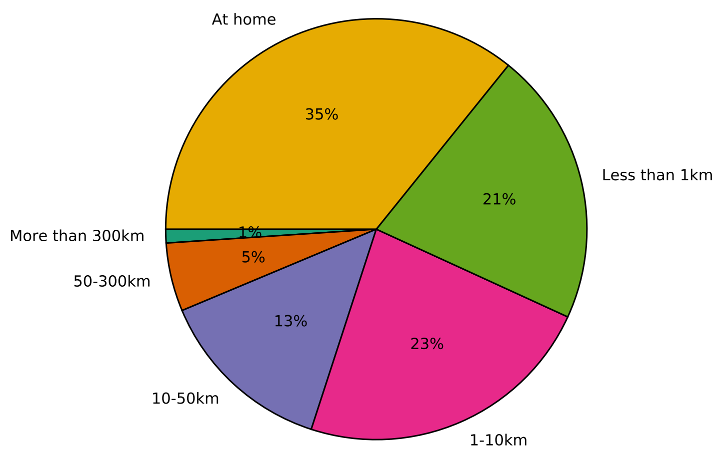


Figure 3.5: Distance between the caregivers and the elders

3.3 Daily Life Of Elderly People

The following paragraphs highlight difficulties met by elderly people in performing activities of daily life. These insights should guide the choice of services when designing an AAL system.

The survey confronts the answers of caregivers with the ones of elderly people. The interest is thus twofold:

- Get information on elderly people' life;
- Estimate the objectivity of elderly people, comparing their answers to the caregivers' ones.

3.3.1 Main Activities Of Elderly People

Figure 3.6 shows the dominance of television and reading in most elderly people' life. Leisure, social and outside activities (at the restaurant, walk, cinema...) also remain important for many elderly people, as well as the practice of some manual activities. These activities should be leveraged by an AAL system (e.g., through service provisioning on a television).

As for outside walk, only a few caregivers have mentioned it, whereas many elderly people consider they do it often. I suppose, with no guarantee, that elderly people are enthusiastic enough about outside walk to mention it as an important activity, whereas the caregivers neglect it. Another explanation might be a delusion from some elderly people, who refuse to admit that walking is no longer part of their daily activities.

However, some elderly people also claim to do nothing of their days. Some of them even highlighted their loneliness and expressed depression in the questionnaire, beyond formal questions.

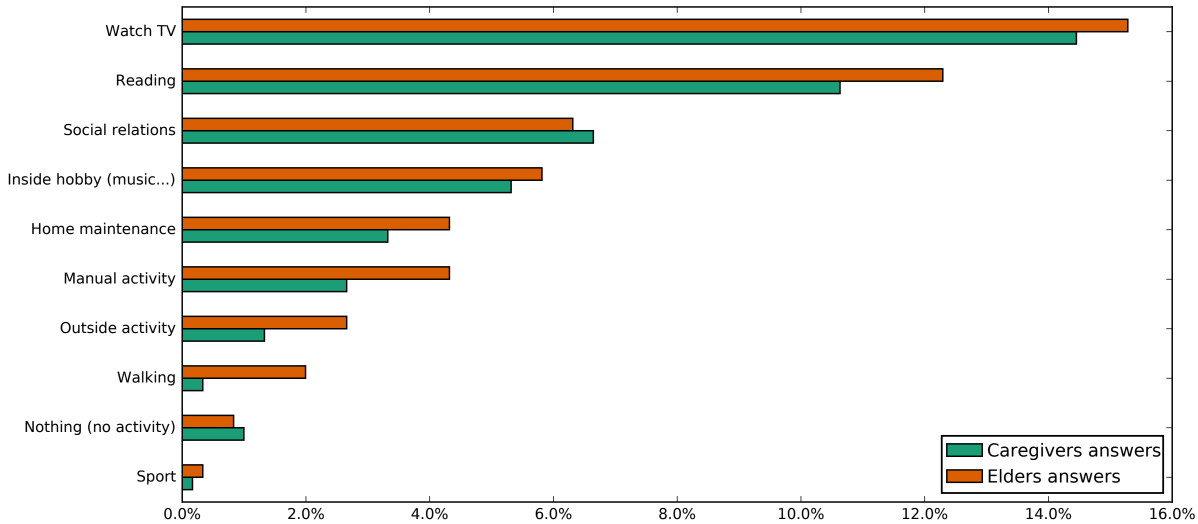


Figure 3.6: Main activities of elderly people

3.3.2 Health Disorders

Technical aids

30% of the surveyed elderly people are already using a remote alarm system (Figure 3.7). We also observe that over 60% of elderly people use a mobility aid (i.e., a cane, a walker or wheelchair). This is an important information about their motor skills. Finally, hearing and visual abilities may be important for the design of AAL services relying on these two senses.

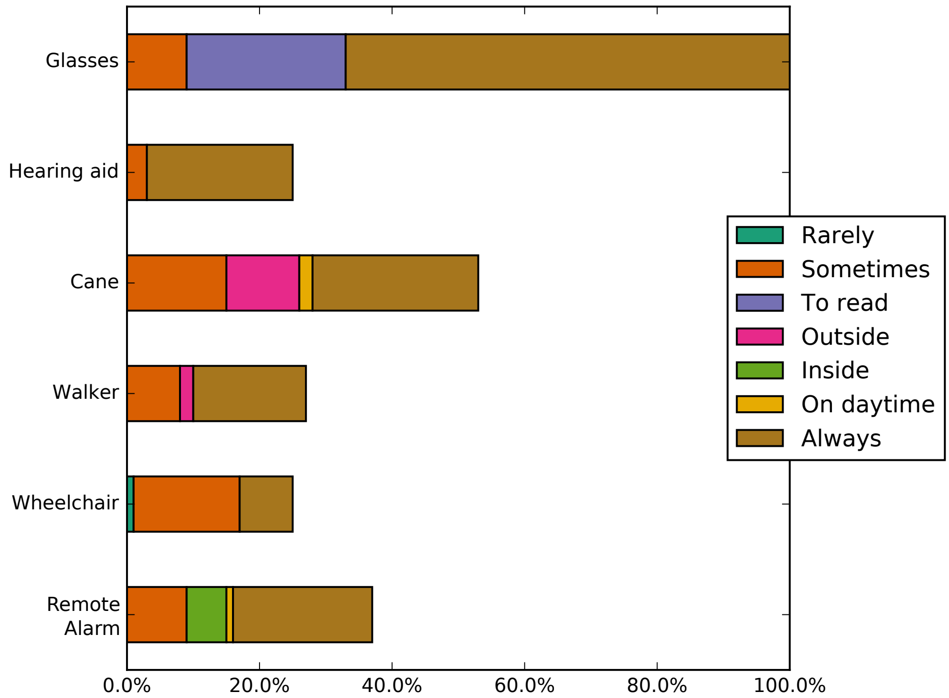
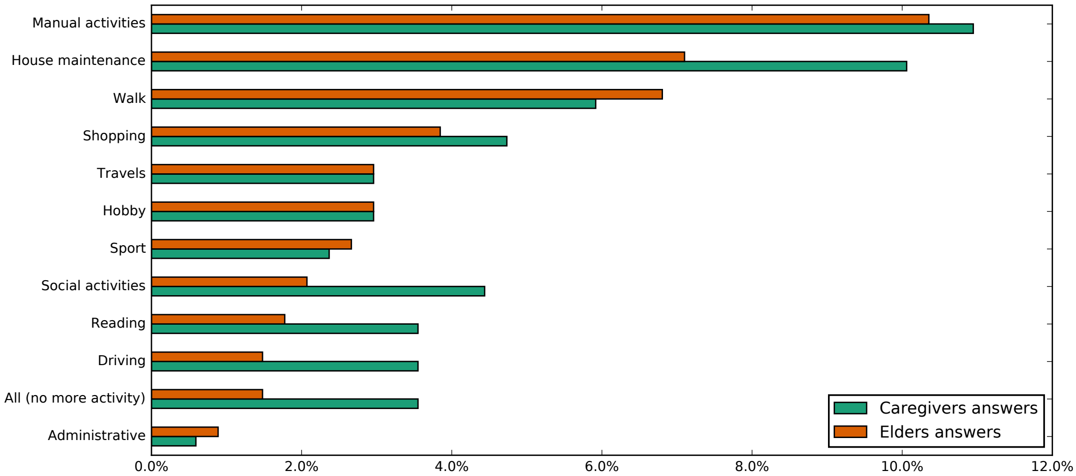


Figure 3.7: Technical aids used by the elders

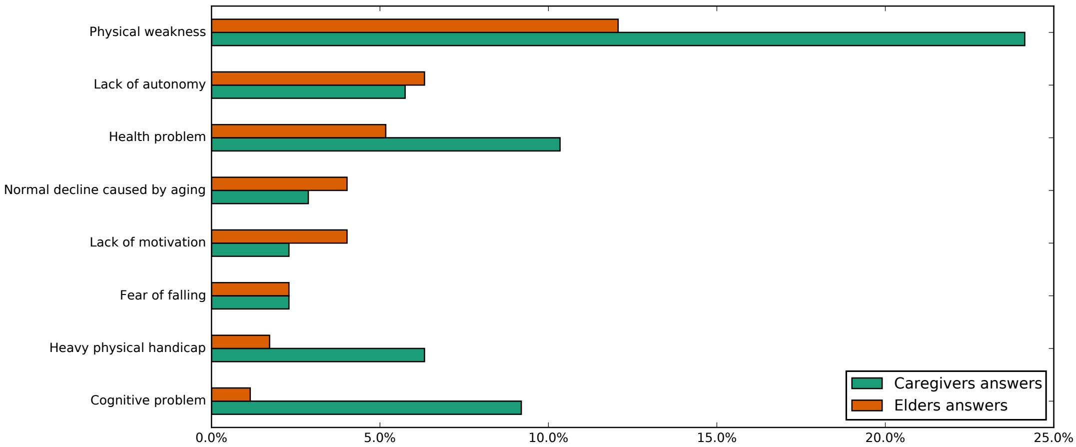
Activities become impossible

Activity limitations are numerous for elderly people (Figure 3.8a). These limitations may be due to physical or moral decline. Some elderly people explain that they are no longer physically able to perform certain activities, whereas others claim a lack of motivation. These activities are either leisure activities (e.g., sport, walk, games, trips) and useful activities (e.g., cleaning, grocery shopping, administrative tasks). We note that caregivers are more effusive than elderly people about activity limitations, certainly due to a denial from these elderly people.

An AAL system should be designed to address lack of autonomy, fear of falling and cognitive problems from elderly people. It might also address motivation issues through serious games, and possibly health problems using specialized services.



(a) Activities became impossible



(b) Causes of activity limitations

Figure 3.8: Limitations of activities

Medication

From Figure 3.9, about 20 to 25% of elderly people are not able to manage their medications or to know when to go to the doctor. Over 50% of elderly people never seem to have problems with their medication.

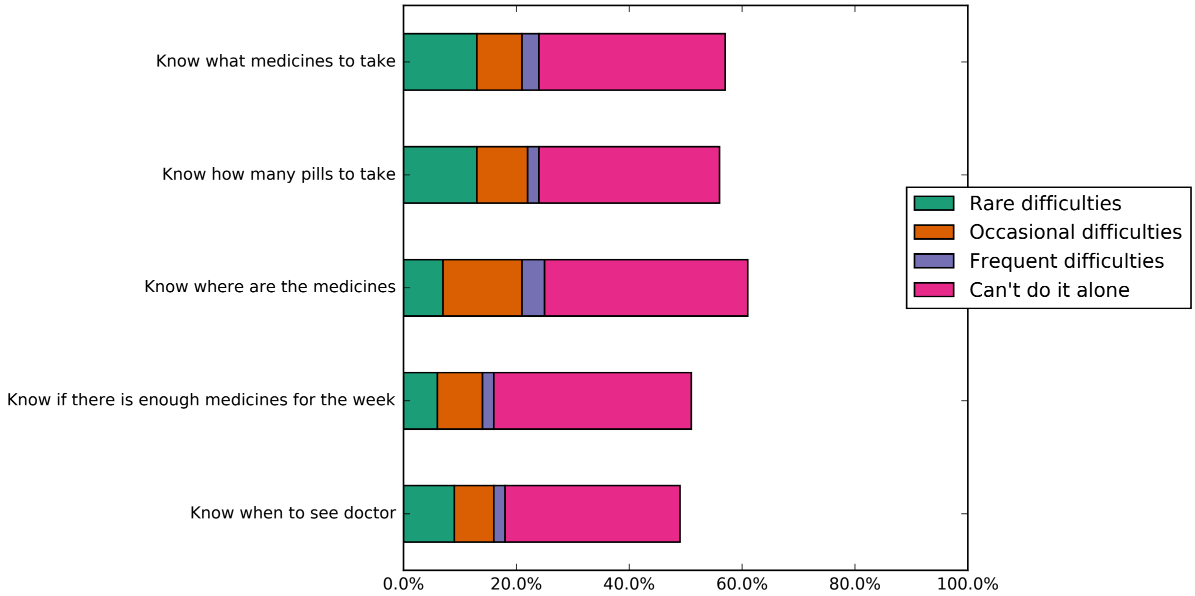


Figure 3.9: Difficulties in taking medications

3.3.3 Daily Activities

Dressing

60% of elderly people have difficulties to dress up (Figure 3.10). Among these, about 20% frequently experience these difficulties, and 20% cannot do it alone.

Eating

Half of the surveyed elderly people have no difficulties to at least heat a dish (Figure 3.11). For those who have difficulties, only half of them obtain a constant help, the others still compose their own meal, despite the difficulties they may experience. Thus, eating is a particularly critical activity for elderly people.

House cleaning

House cleaning is a challenging activity for elderly people (Figure 3.12). Indeed, 30% of the surveyed persons say they have difficulty using the broom and clean the dust, and another 30% never do it alone. There are also frequent difficulties in moving between rooms, and wandering is a well known problem for people suffering from Alzheimer.

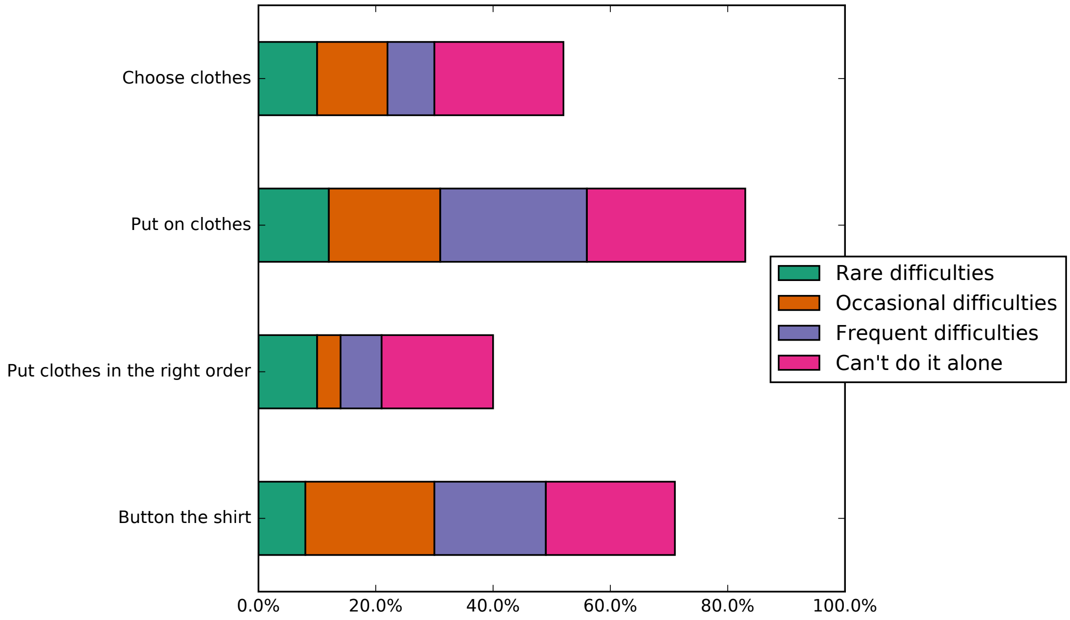


Figure 3.10: Difficulties in dressing

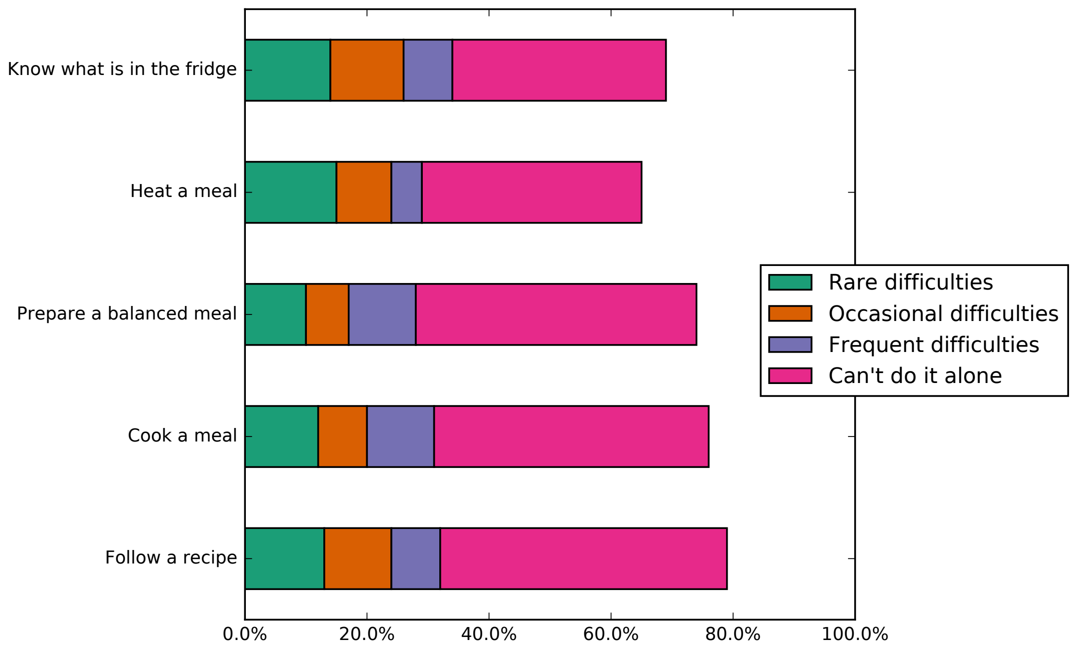


Figure 3.11: Difficulties in eating

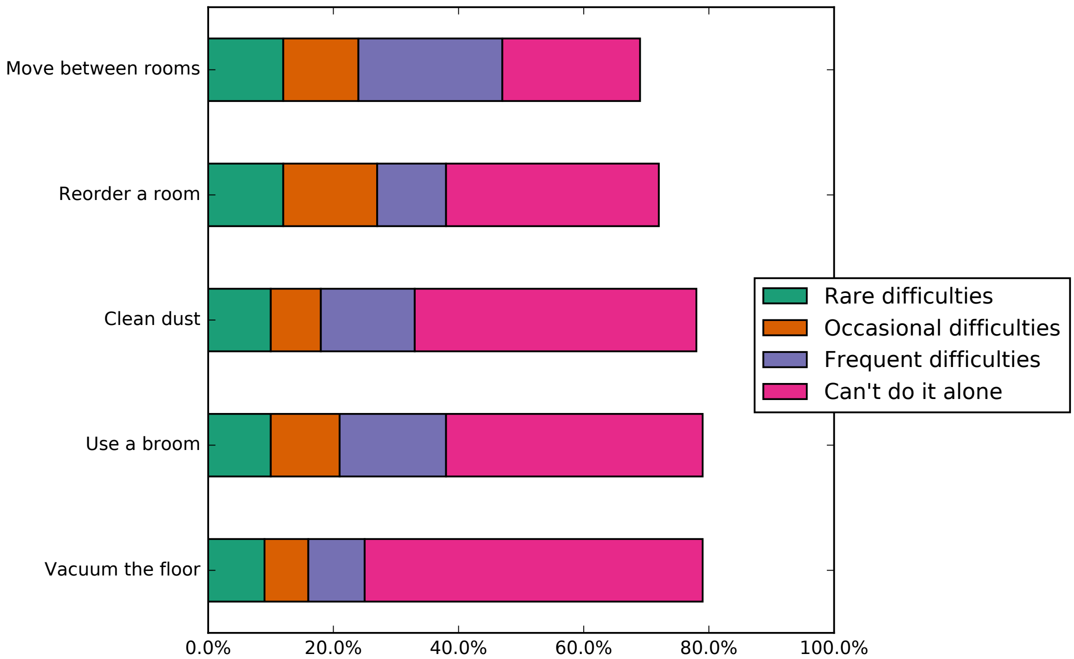


Figure 3.12: Difficulties in cleaning the house

Administrative Activities

Figure 3.13 is divided into three categories: elders who experience no difficulty managing administrative tasks and finance (40%), those who have difficulty more or less frequent (about 30%) and those who do not alone (25 to 40%).

Home appliances

Among the home appliances used by elderly people (Figure 3.14), we observe once more the importance of television. It is owned and used by at least 80% of elderly people. Assistive services for elderly people could thus take advantage of television. In contrast, less 30% of elderly people using a computer and the Internet.

The home phone is used by most elderly people, and 30% of them own a mobile phone (although only half of them use it). As for the kitchen tools, the vast majority of elderly people use oven and fridge. Kitchen accessories (toaster, coffee maker, kettle) are also used by 40 to 60% of elderly people. These figures are reassuring, because they show that despite some difficulties observed previously, a large number of elderly people continue to use most daily life objects.

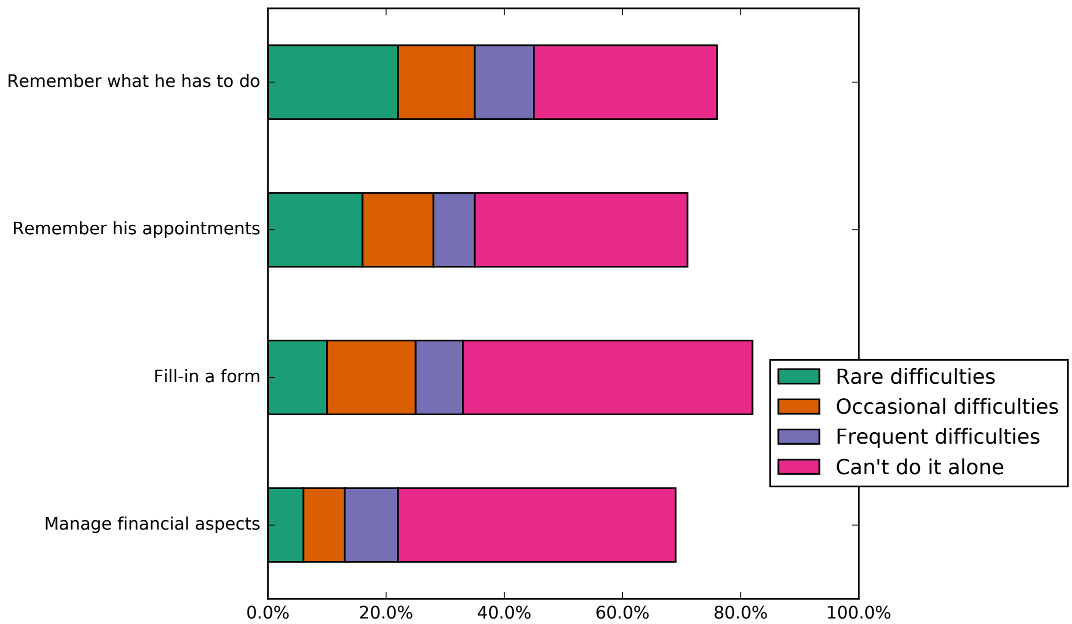


Figure 3.13: Difficulties in performing administrative activities

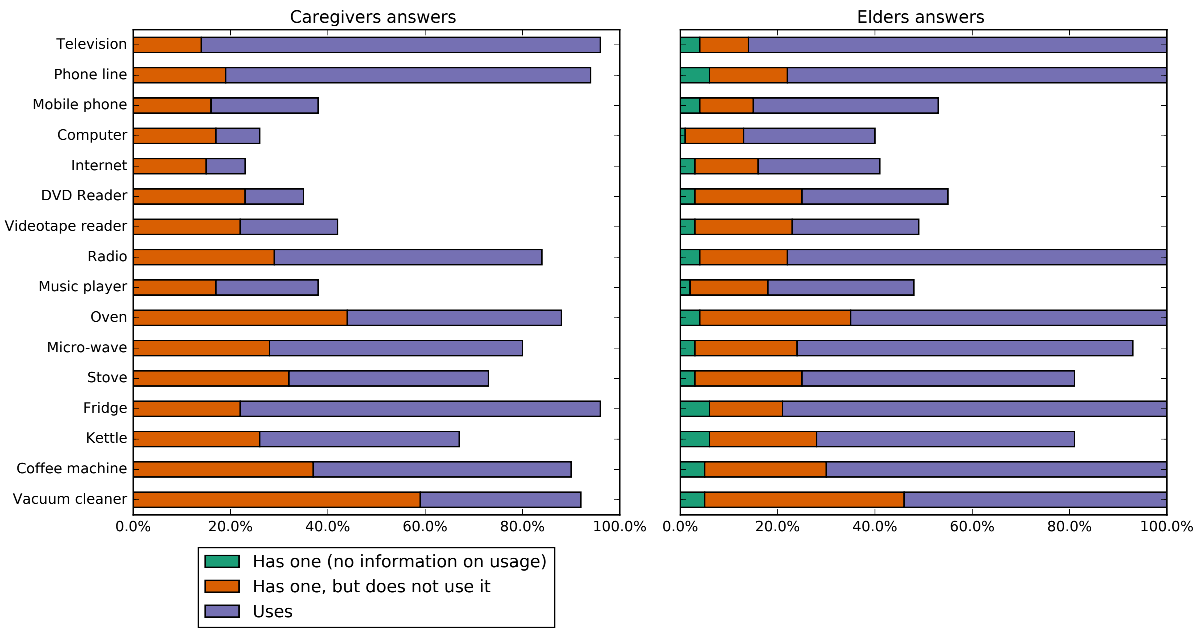


Figure 3.14: Products of daily life used by the elders

3.3.4 Contact With The Outside And Security

Outdoor activities

Figure 3.15 once again reveals a frequent pattern in this survey: 60% of elderly people surveyed have outdoor activities, and 40% do not. This proportion has been observed in most of the questions regarding the difficulties experienced by elderly people in their daily activities. It is certainly biased by the fact that some elderly people and caregivers refrain from answering certain questions, but we can assume the existence of two types of people among the elderly population:

– Persons who continue to have a rather normal life (50 to 60% of the surveyed population);

– Persons who are largely dependent (40 to 50% of the surveyed population).

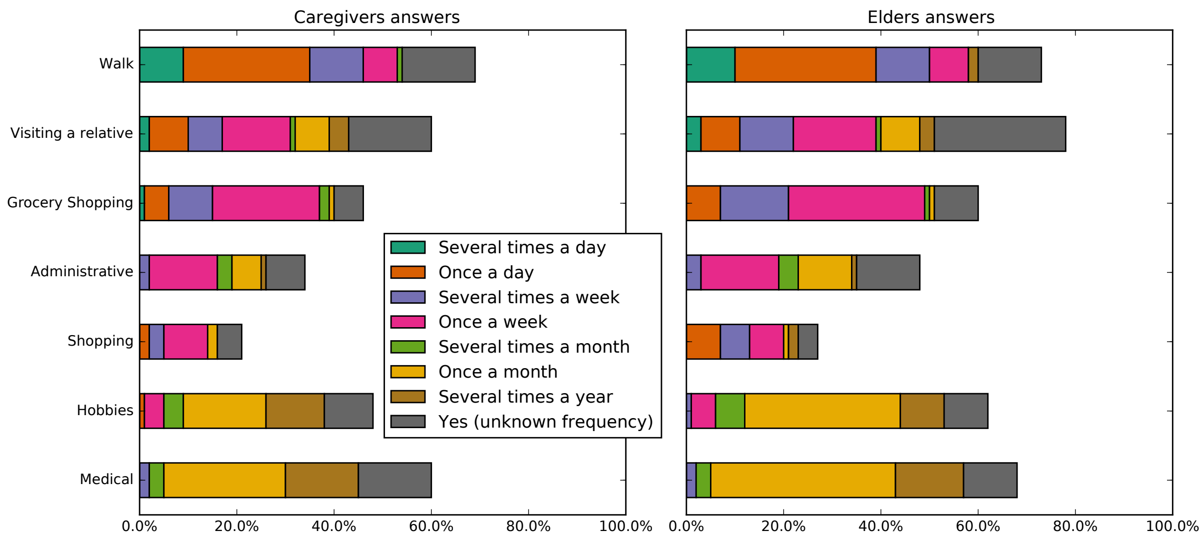


Figure 3.15: Outdoor activities performed by the elders

Not surprisingly, among elderly people who practice outdoor activities, we observe daily activities (walk), weekly activities (e.g., grocery shopping, visits to a relative), monthly activities (go to the doctor) or occasional activities (administrative, for a limited number of people). It is nonetheless interesting to note that for most elderly people, recreative activities (e.g., cinema and restaurant) appear to be rare, and elderly people answers are often in contradiction with the ones of the caregivers.

Transportations

According to Figure 3.16, 60% of elderly people use a relative's car to move around. In addition, 30% of elderly people sometimes use an ambulance. We also note that 30%

are still capable of driving their own car. Walking is also used (60% of elderly people), although more rarely the car. Finally, public transportations are very rarely used.

These informations could be leveraged when considering the extension of an AAL system from smart home towards smart city.

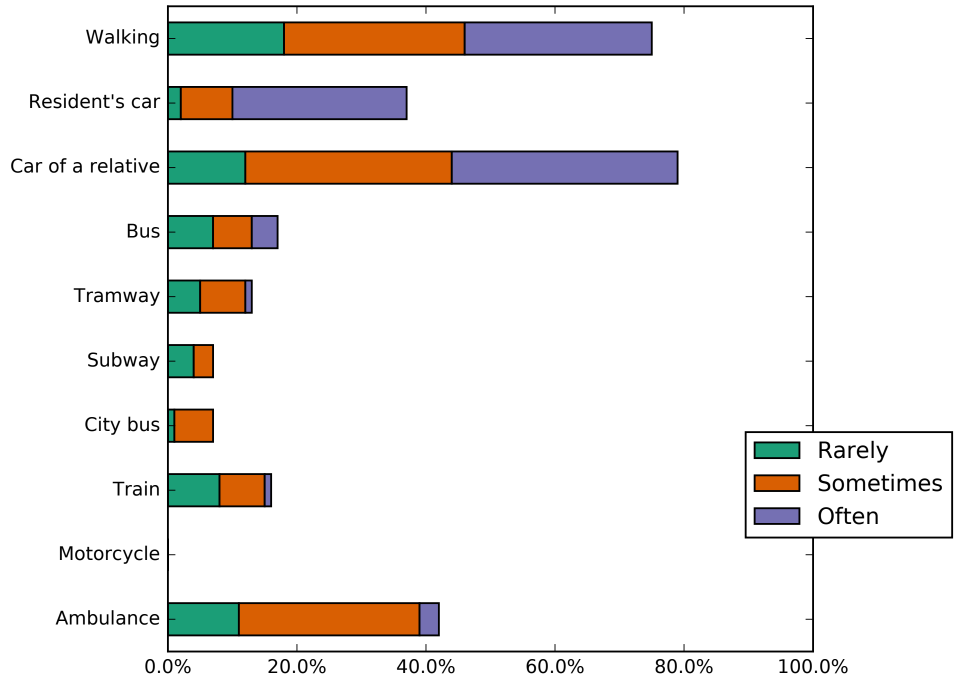


Figure 3.16: Transportations used by the elders

Using a phone

Using a phone seems to be a problem for about 50% of elderly people (Figure 3.17). Many of them have difficulties to hear and follow a conversation or even to know the identity of the interlocutor. 10% are not even able to use the phone. This information is important before considering services that include call centers.

Feeling of security at home

The vast majority of elderly people (95%) feels secure in their home (Figure 3.18). Only 5% seems to develop a feeling of constant oppression, or a fear of being disturbed by an unknown visitors, which is often symptomatic of depression. As for the caregivers' opinion, 10% do not think that the elderly person is safe, but the reasons are very different. Caregivers are concerned about the elderly people's health, especially the fear of a accident, and a general lack of autonomy. It is surprising, however, that among the elderly people themselves, few seem concerned by the risk of having an accident in their home.

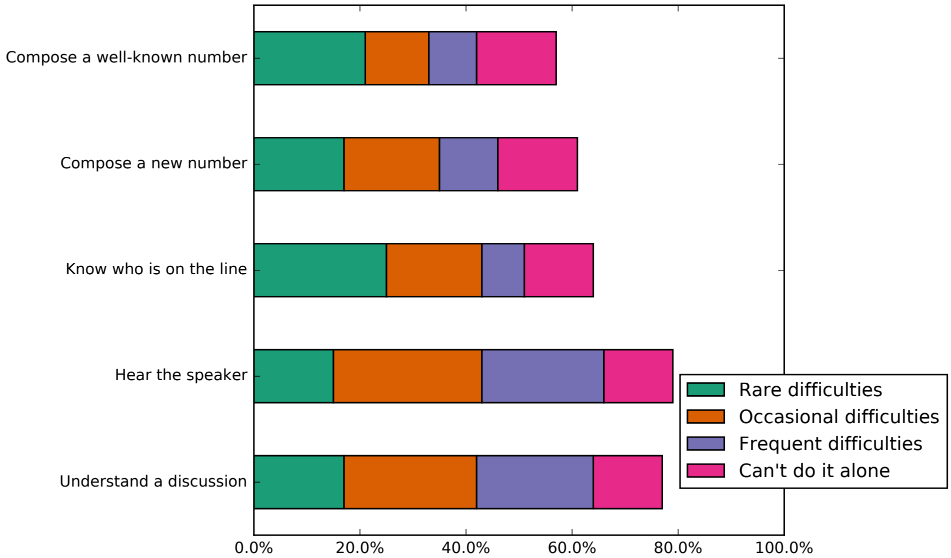


Figure 3.17: Difficulties in using phone

Feeling of security outside

Figure 3.19 shows that 75% of elderly people feel safe outside, against 70% of caregivers. The main cause of insecurity is by far the risk of fall. In general, both elderly people and caregivers are affected by the elderly people's physical and cognitive problems. It should however be noted that this question may be biased. Some persons feel safe because they never go out alone, and others do not go out anymore, but still answered the question.

3.4 Aids Received By Elderly People

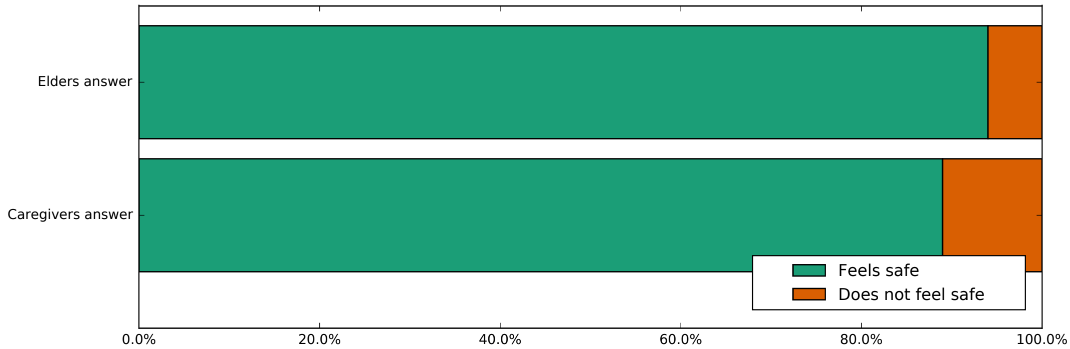
Monitoring services

A third of the elderly people already have a monitoring service (according to caregivers), the most common being a call center (Figure 3.20). An AAL system could make use of these existing services (e.g., by integrating them into the system).

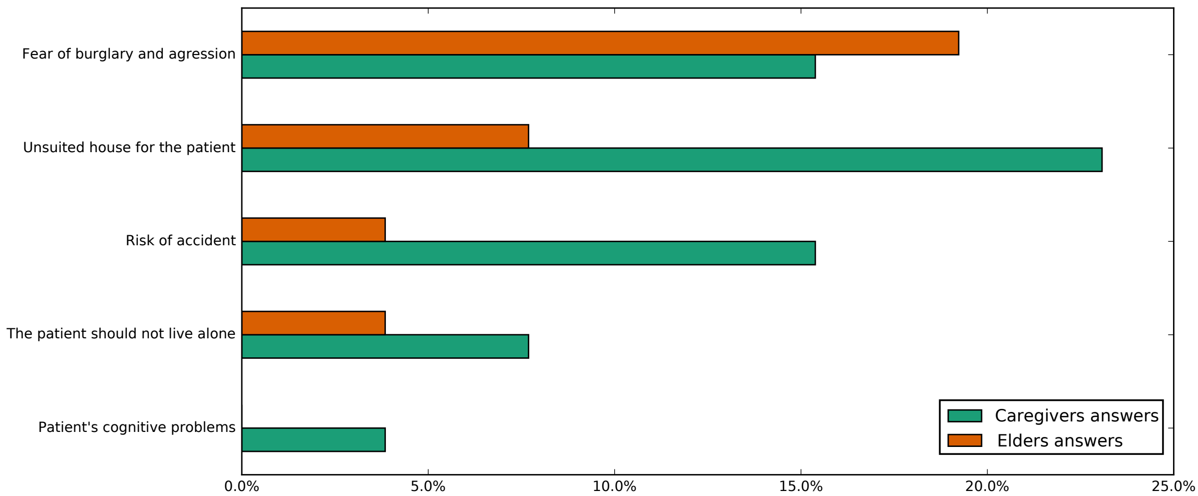
Home-Care Services

Generally speaking, approximately 60% of elderly people are using home-care services (Figure 3.21), in a frequency ranging between once a day and once a week. 30% to 40% of elderly people use a laundry service or home maintenance service at least once a week. Meal delivery and grocery delivery services are quite rare, as they are used by less than 30% of elderly people, usually weekly or daily. 50% of elderly people also employ a nurse or medical carer; either daily or monthly depending on the elderly people's needs.

An AAL system could interact with home-care services, by asking the professionals to

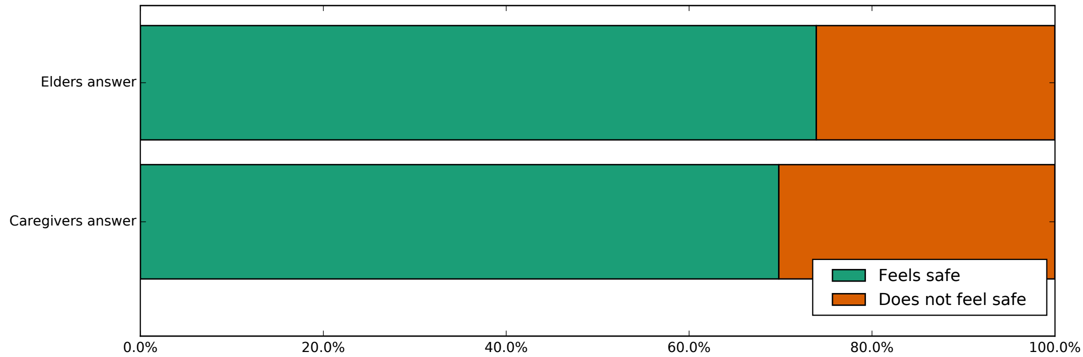


(a) Feeling of security inside elderly people' home

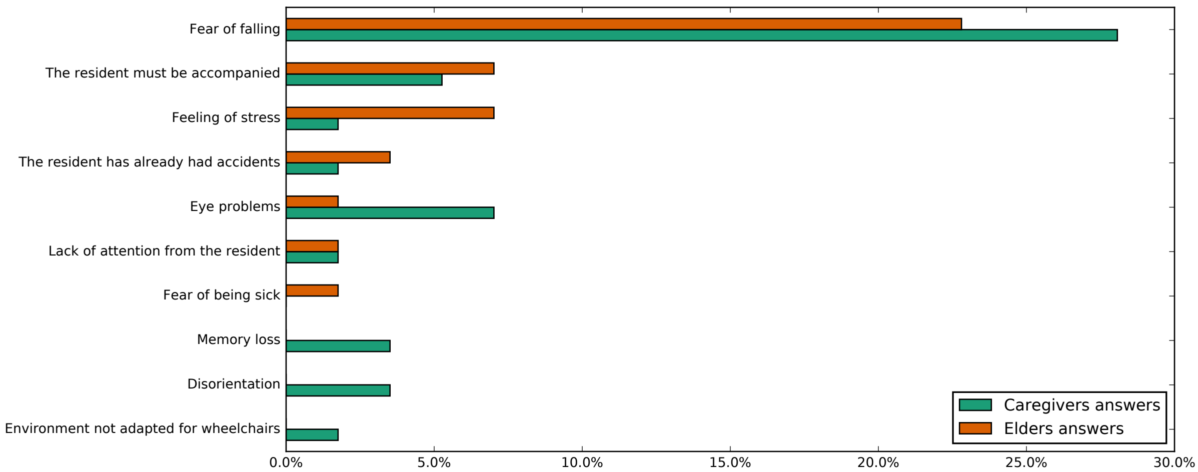


(b) Causes of insecurity inside the house

Figure 3.18: Feeling of security inside elderly people' home



(a) Feeling of security in the outside



(b) Causes of insecurity in the outside

Figure 3.19: Feeling of security in the outside

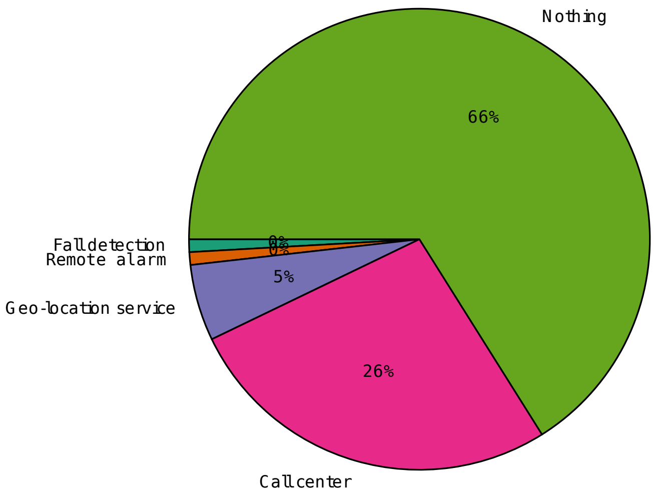


Figure 3.20: Monitoring services used by the elders

provide qualitative information about the elderly person, or optimizing the frequency of visits.

Opinion Seniors on home-care services

Figures 3.22 help us understand the opinion of elderly people and caregivers towards home-care services. We observe that for many elderly persons it is important to stay at their home, or to have a daily presence to talk to. Of course helping the elderly person is the primary role of these services, but they also support the family caregivers physically and morally, and contribute to a feeling of security.

However, several persons are frustrated with these services. They feel embarrassed by the irregular schedules of some professional caregivers, which disrupts their activities. Moreover, the presence of a external person, although appreciated by some, is also considered as a loss of freedom and discomfort by others. Finally, many elderly people criticized the lack of training of caregivers, and the fact that caregivers change too often to create a real connection. We must recall also that the work of professional caregivers is extremely difficult.

Finally, when asked about the criteria for choosing a home-care service, elderly people mainly put forward the professionalism in all its forms: serious, punctuality, efficiency, discretion, patience, understanding and above all, trust. The frequency and availability are also cited. We note that the cost is only slightly mentioned. On the caregivers side, a large number of caregivers are enthusiastic about home-care services, and pay attention to professionalism, quality of service, and to the human qualities of the professional caregivers. However, we also observe that special attention is paid to the scheduling of these services, which they want regular and flexible. Finally, many caregivers appreciate that home-care services improve the quality of life of elderly people (e.g., better hygiene,

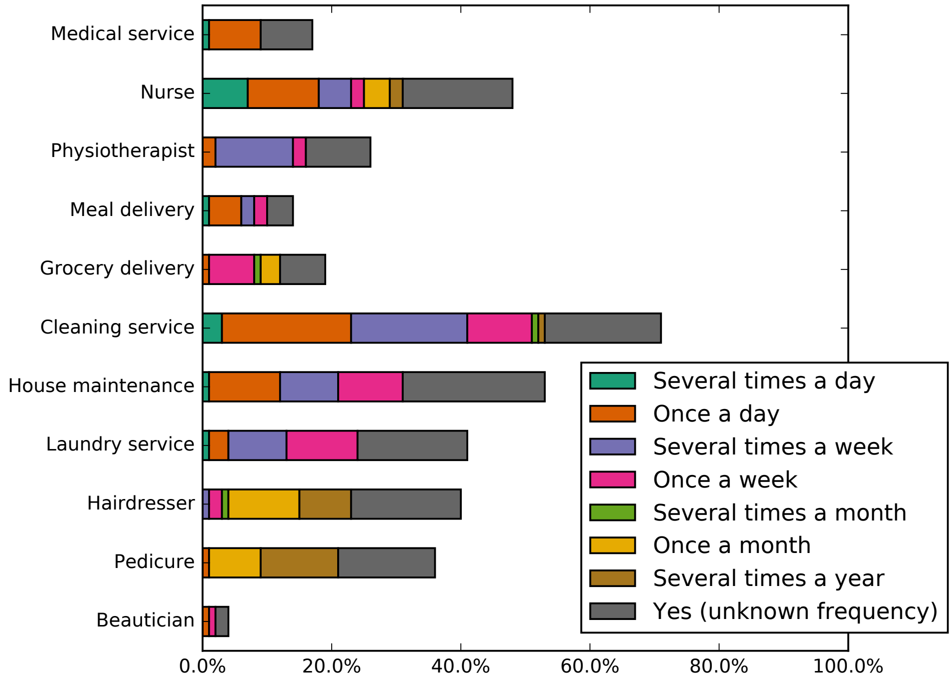


Figure 3.21: Home-care services

balanced food)

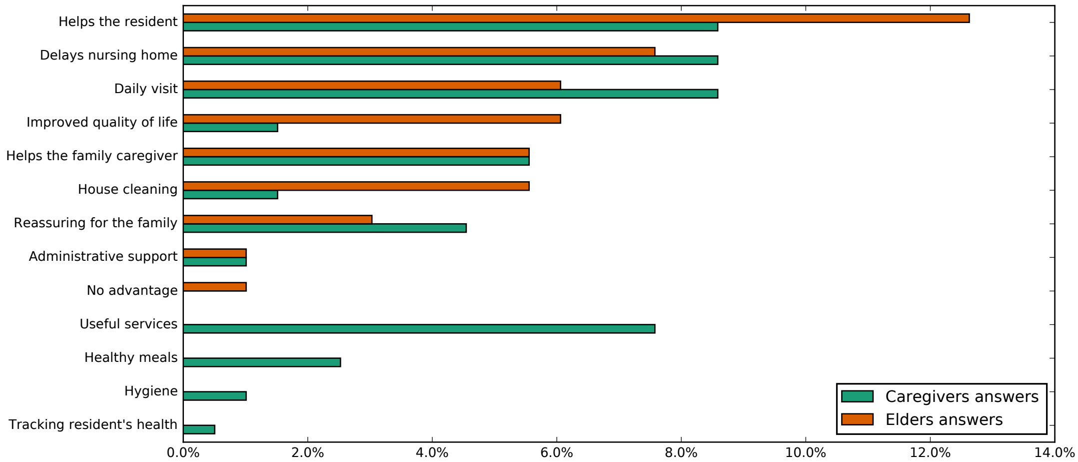
Financial helps

From Figure 3.23, the majority of elderly people receive between 0 and 40 hours aids home per month. Yet, over 30% claim that not to be entitled to any financial assistance, and charge any service on their own budget.

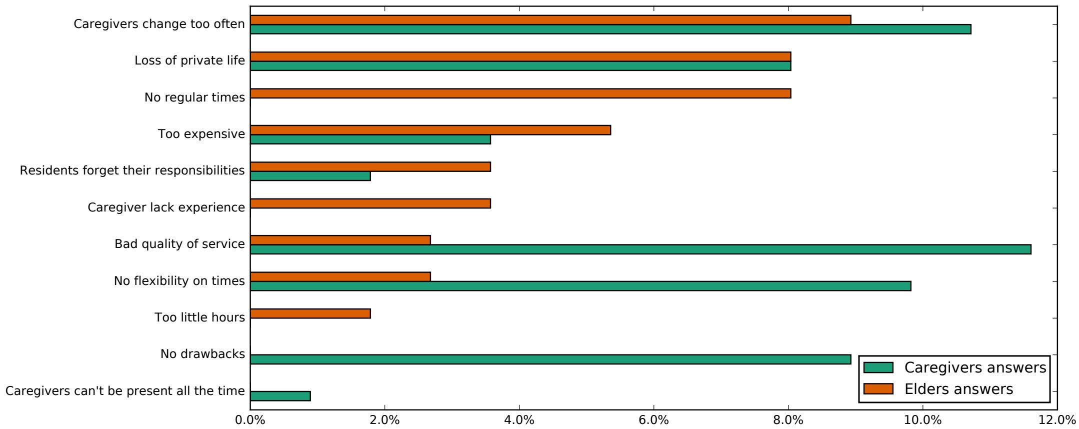
3.5 Openings

3.5.1 Search For Parameters Of Importance

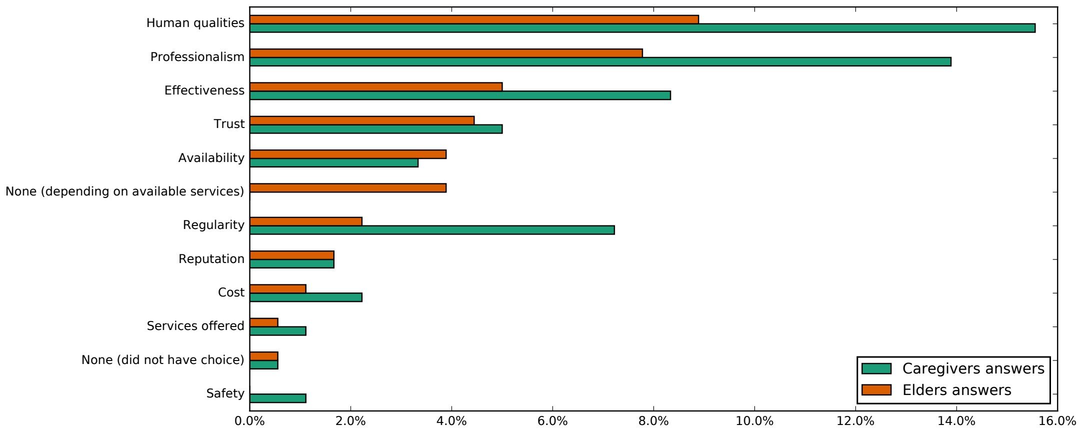
Based on this study, the elderly people population seems to be divided into two categories: persons who are immobilized by old age and/or cognitive decline and have a high dependence, and those who continue to live a normal life. It would be interesting to identify the questions which are most discriminant of these categories, in order to obtain an indicator of dependence for elderly people and adapt the services they receive accordingly.



(a) Advantages



(b) Drawbacks



(c) Criteria for choosing home services

Figure 3.22: Elderly people's opinion about home services

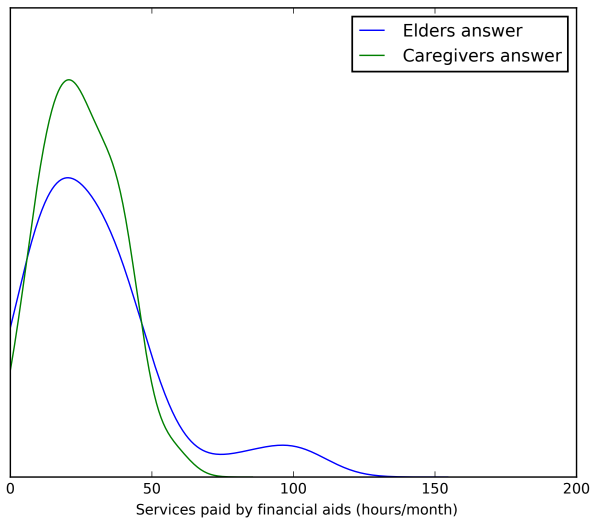


Figure 3.23: Financial helps received by elderly people

3.6 Synthesis — The Needs Of Elderly People

3.6.1 Expectations Of Elderly People

Few elderly people responded to the open question about their expectation (Figure 3.24). In contrast, the caregivers have expressed numerous expectations towards additional aid. These responses include enhanced home-care services (more caregivers, better medical support), or new services (remote monitoring, support when going out). Some caregivers stressed the importance of better medical equipment. A few others also wished to receive moral support for themselves. A memory workshop was suggested by several elderly people and caregivers. This could be considered, by integrating serious games to AAL.

3.7 Analysing Differences Between Elderly People And Their Caregivers

This analysis allows to estimate the differences in perception between caregiver and elder. To understand these divergences, I distinguish two types of questions: objective questions, and subjective questions.

Objective questions concern quantifiable facts, such as the elderly people's habits. Reassuringly, the responses of elderly people are generally identical to those of caregivers in these questions. Sometimes elderly people seem to overestimate their abilities, considering for example that they often take the car, or that they rarely experience difficulties in an activity, while caregivers say that these persons rarely walk and have often difficulties in

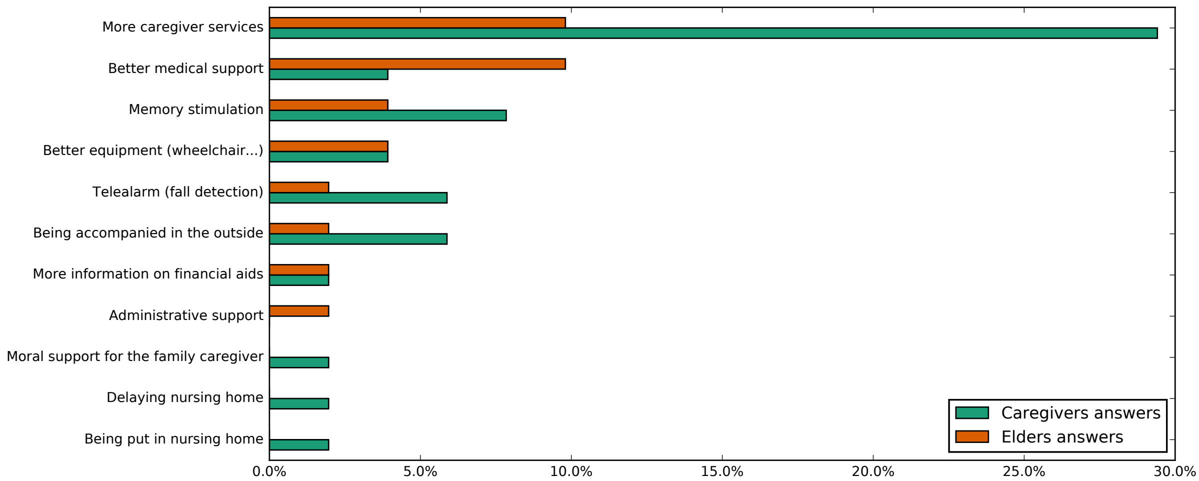


Figure 3.24: Services requested by the elders

performing an activity. Nevertheless, it remains seldom that the responses from an elderly person and from his/her caregiver are opposed.

On the other hand, subjective questions depend on the elderly person' and caregivers' opinions regarding elderly person' situation. It includes opinions on home-care services, on the elderly people's safety, and on the causes of limitations in their activities. Among subjective questions, yes/no questions usually get similar answers (i.e., less than 10% divergence), for instance when it comes to whether the elderly person is safe. Open questions however cause more divergences; caregivers often give further analysis of the situation, and elderly people rarely mention their own health problems.

Generally speaking, this analysis is reassuring, both on the awareness of elderly people regarding their situation, and on the understanding of caregivers about the persons they support.

3.8 Conclusion

Through the different insights displayed in this chapter, multiple user needs have emerged. I summarize these user needs as follows:

- Ensuring the elderly people's safety;
- Supporting caregivers;
- Improving the perception of home-care services.
    - Providing a more regular staff in home-care services.
- Following the evolution of Alzheimer;
    - Adapting to the elderly people's profile and the evolution of the disease.

Certainly, to stabilize professional caregivers and support family caregivers, it is necessary to simplify their work and reassure them towards elderly people' health. An AAL system could guarantee the safety of the elderly person, or alert the caregivers in case of risk, thus reducing their stress. A good management of the information from this system could also facilitate the work of caregivers and make them more effective.

This study suggests the importance of monitoring and coaching elderly people outside, when they go out for a walk or for grocery shopping. Indeed, this aspect of their lives is hardly supported by professional caregivers, and a large number of elderly people say they have physical weaknesses, which makes them vulnerable to a fall.

This study also highlights some specific points that need to be taken into account by an AAL solution:

- Television is the best technology to interact with elderly people;
- A majority of caregivers are elderly people themselves;
- Every elderly people have widely different profiles and needs;
- The use of the phone (call center) is not possible with every elderly people;
- A number of elderly people often go out for a walk or to by groceries;
- Loneliness is a major problem, but seems to affect relatively few elderly people;
- Many elderly people feel uneasy with administrative tasks;
- Eating meals and taking medication appear to be the two main problems of everyday life;
- Elderly people are aware of the importance of home-care services, but are often frustrated by the loss of freedom;
- More than half of elderly people are living alone.

Following in this dissertation, I describe the UbiSMART platform for AAL, that we have developed in response to these observed user-needs.

Part III

A Framework for Ambient Assisted Living

He who works with his hands is a laborer.
He who works with his hands and his head is a craftsman.
He who works with his hands and his head and his heart is an artist.

Francis of Assisi

4 Addressing Technical Challenges

4.1 Introduction

4.1.1 Objective: Large Scale Deployment

The clear purpose of this thesis is to provide a systematic method for optimizing the reliability of Activity Recognition. Chapters 6 and 7 describe this method step-by-step. Yet, this thesis takes place in a wider scope, that is to perform large-scale deployments of an AAL system in real situations, so that it can benefit to real users.

Arguably, the best way to improve a system is to gather feedbacks, and to learn from these feedbacks. Thus, my team and I benefit from the deployments we perform, in the sense that it brings a feedback about our system, UbiSMART Although I do not reject computer simulations nor laboratory experiments, I argue that an AAL system should primarily be designed towards real deployments. Real deployments introduce multiple technical constraints that are not found during prototyping. It may sometimes bring one away from one's comfort zone, but I consider it to be a necessary cost for the research in AAL to move forward. Only through real deployments will we understand the real needs of the elderly people, and think of pragmatic solutions. With this in mind, and after several years of work from several contributors, my team and I have been able to perform several real-world deployments.

In this chapter, I describe several functional requirements of an AAL framework, and the solutions we propose with UbiSMART. Most of these are practical solutions, and they contain only little theoretical concerns. However, I believe that in an applied research like AAL, technical contributions are necessary as well as theoretical ones, and both deserve to be reported, so they can enrich the literature and carry the research forward. Figure 4.1 gives a complete view of UbiSMART.

Rather than listing every single components of UbiSMART, I prefer to introduce more context, by adopting a chronological perspective. Section 4.2 describes the original UbiSMART framework, that was available when I began my thesis work. Section 4.3 highlights the many problematics that have emerged from our early experience in deploying UbiSMART in its original design. Section 4.4 explains how we addressed these problematics,

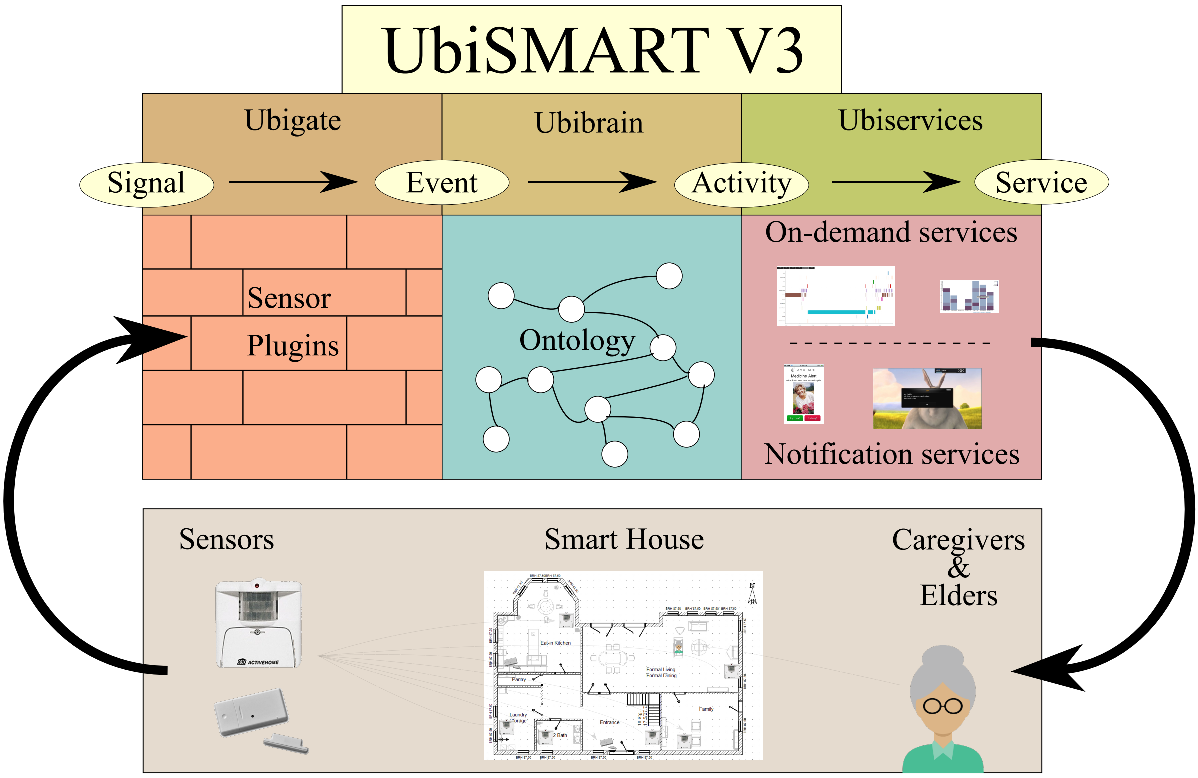


Figure 4.1: Design of the UbiSMART framework

and what is the current version of UbiSMART through these improvements. To maintain the integrity of this dissertation, I will pay a special attention in isolating my own contribution from my colleagues contributions. Finally, in Section 4.5 I discuss the outcomes, and the lessons we have learned from our deployments. I introduce several perspectives, that can illustrate potential evolutions of UbiSMART as a framework for AAL and pervasive computing.

Sections 4.3 and 4.4 are complementary, and share similarities in their structures. The former describes the problematics, where the latter explains our solution. In my opinion, the two of them are equally important, as a good solution can only emerge from a well defined problematic.

4.2 Initial Situation

The UbiSMART framework is the consequence of the work from many researchers over the years, directly or indirectly. But the main credit comes to Aloulou and Tiberghien, who originally designed and implemented it during their thesis work. I had the chance to contribute to it as well, during my master degree internship. They created it as a wrapper around their reasoning engine. The goal was to provide a complete AAL framework, rather than solely executing the reasoner over data and providing a result as output. Its initial design came from the observed need to evolve, from a proof-of-concept to a proof-of-value. Whereas the reasoning engine and basic service delivery were originally an interesting proof-of-concept, their practical value was very limited. Thus came the idea of wrapping it in a more complete framework for AAL, that would impact real users.

To mention the original UbiSMART framework from 2012, I will use the name UbiSMART V2 (UbiSMART V1 was solely a prototype, and will not be mentioned here). Every subsequent work described in this chapter will take place in UbiSMART V3, called UbiSMART by default.

Figure 4.2 introduces the design of UbiSMART V2, as it was introduced in Tiberghien's thesis [31]. Despites the complexity of this diagram, we can see a flow emerging, from sensor data to service delivery, with reasoning as the intermediate process. It is important to note that this architecture was designed before the implementation, and that it has never been fully-implemented in UbiSMART V2. Nevertheless, much of these concepts have remained as foundations for UbiSMART V3.

4.2.1 UbiSMART v2 Implementation

UbiSMART V2 has been developed using Service-Oriented Architecture (SOA) to increase its modularity. More precisely, it was developed using Java and the OSGi framework for SOA [1]. It was composed of several modules, appearing in Table 4.1. Each module is developed independently, and communicates with the others through well-defined inputs and outputs.

4.2.2 UbiSMART v2 — Use Case

The development of UbiSMART V2 had been motivated by two deployments, named AMUPADH and Quality of Life (QoL).

The AMUPADH project included a deployment in the PeaceHaven Nursing Home, in Singapore. It was made in collaboration with other research teams, which focused on the infrastructure and deployment, while the UbiSMART team focused solely in recognizing activities.

The QoL project started with three houses being deployed in France. These houses were deployed by an external company. Here also, the UbiSMART team only had to focus in recognizing activities.

These two deployments relied on external data providers, and the role of UbiSMART was restricted to activity recognition. Therefore, the implementation of UbiSMART V2 is centered towards real-time reception of data, reasoning, and providing an output. A wider scope had been included into the design, with data acquisition and service provisioning, but they remained mostly abstract, and were not properly implemented. Moreover, the UbiSMART V2 developers had had little contact with the end-users at the moment.

In 2012, early in my thesis, a real deployments was considered, where UbiSMART developers would be responsible of the complete deployment, and would keep a strong

[1] http://www.osgi.org/Main/HomePage

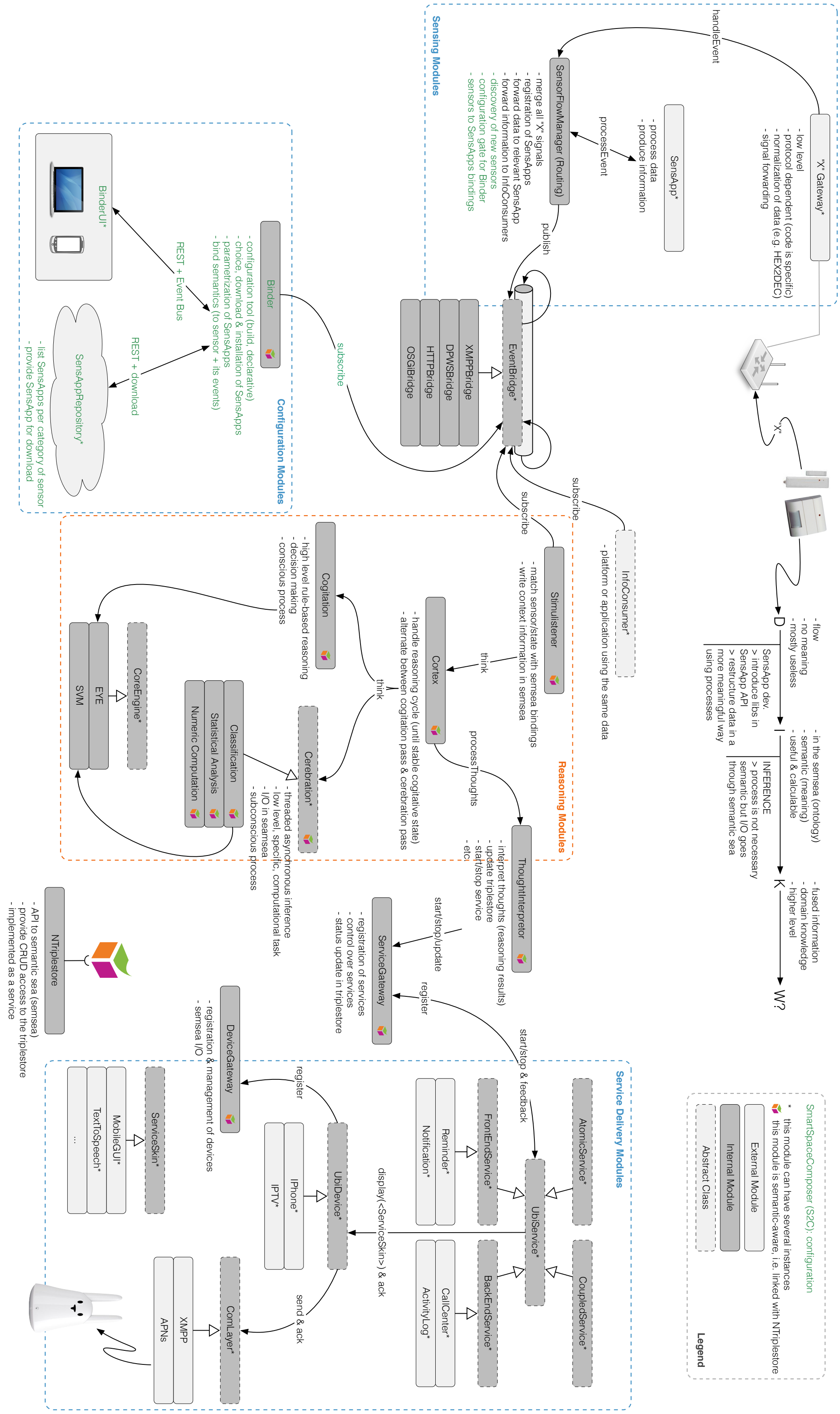


Figure 4.2: UbismartV2

Modules	**Description**
DatabaseReader	Read data from the database
DatabaseGateway	Sends collected data from the database through the *OsgiBridge*, to the *SensorFlow manager*
SensorGateway	The Gateway for the sensors deployed in the house
VirtualClock	A virtual sensor that sends the current date and time in a regular time slot
VirtualSensor-Gateway	The Gateway for virtual sensors such as the *VirtualClock*
OSGIBridge	This bundle handles the *OSGi Event* communication between bundles
SensorFlowMan-ager	Receives events from the *gateways* and forwards them to *Stimulistener*
Stimulistener	Updates the semantic model with external information, and launches the reasoning process
Cortex	Handles the *Cogitation* and *Cerebration* reasoning
Cogitation	Uses the EYE reasoner to perform semantic reasoning using defined semantic rules
Cerebration-MotionEstimator	Used to calculate the activity of the patient in each room of the house during a window of time
Cerebration-Evasion	Controls if the patient runs away from his house
Cerebration-DSTDecisionMaking	The Dempster-Shaffer decision making bundle, used to take decision in conflicting and uncertain situations [30]
SensApp-MassFunction	Used by Dempster-Shaffer to calculate MassFunction values
SensApp-FunctionalParam	Used by Dempster-Shaffer to calculate a functional parameter
Service-ReminderService	A service allowing the send reminders on tablets and smartphones
EYEReasoner	A technical module to launch the EYE reasoner
N3TripleStore	A technical module to read/write/update N3 files
Cerebration-abstractClass	A template for the different cerebrations bundles
EventBridge-abstractClass	A template for the bundles handling communication between two bundles
Toolbox	Contains some tools for reading config file, logging, time parsing, etc.
ClusterLogger	A centralized logger.

Table 4.1: Technical architecture of UbiSMART V2

contact with the end-users. As the project materialized, we saw several limitations and questions appearing from UbiSMART V2. These limitations are listed extensively in the following Section 4.3.

4.3 Problematics

As explained in Section 4.2, UbiSMART V2 was not created with full-stack deployments in mind. There was no implementation for sensor acquisition, and the practical steps for deployment were out of its scope. Its output had a value for research, but could not benefit to a end-user, as it only consisted of log files.

Through this section, I describe more extensively the questions raised during the shift from UbiSMART V2 towards a full-stack framework for smart-home. I will then answer every of these questions in Section 4.4, by describing the changes we have made in UbiSMART V3.

4.3.1 How To Set-Up A Smart-Home?

The first question one may ask when creating a smart home is "what does a smart home looks like?".

There are two possible types of architectures for smart home: centralized or decentralized. In a centralized architecture, data from the smart home are uploaded to the Internet, and processed on a remote server. On the other hand, a decentralized architecture will handle all the process inside the home, without requiring a connection to the Internet. UbiSMART V2 did not handle data acquisition, thus could equally be adapted to work in a centralized or decentralized architecture. A centralized architecture requires at least a computer used as gateway, physically placed inside the home to forward sensor events to the remote server. It also requires an Internet connection. A decentralized architecture requires a computer too, on which the whole framework will be installed.

Centralized architecture have a drawback, because they depend strongly on the Internet. But compared to decentralized architecture, they are easier to maintain and to debug remotely. Also, the range of decentralized architectures is limited to the home, meaning that services can hardly be accessed from the outside. This is problematic when trying to reach caregivers. Taking all these aspects into account, we have chosen to use a centralized architecture, more powerful and more reactive to a quick development cycle.

The installation process also must be taken into account. The setup should be as simple as possible, and every complex tasks should happen before the deployment, not during. A useful metric is the time spent at the person's house to make a deployment. This problematic has been tackled in Section 4.4.3.

4.3.2 How To Connect The Smart-Home To The Server?

Since we have opted for a centralized approach, we need the smart-home to communicate with the server.

The most critical issue is to obtain an Internet connection. This could seem obvious, but in elderly people's house, it is not usual to have an Internet connection available through Wifi or Ethernet. Section 4.4.1 will show how the use of 3G networks attempts to solve this problem.

Once the Internet connection is established, we need to define the communication protocol between the gateway and the server. The communication protocol must verify the following properties:

– Maintaining reliable communications while using an unreliable protocol (3G internet are often slow, offer a limited amount of data, and suffer from numerous network failures);

– As we will handle streams of data, a persistent connection is likely to be more efficient than a stateless connection. This excludes the HTTP/HTTPS protocol;

– Handling bi-directional connections;

– Being secure.

To make the communication reliable even with frequent disconnections, we must assure a quality of service (preferably maintained internally by the communication protocol). Therefore, we prefer the use of a Publish&Subscribe model. Indeed, one server is dealing with several houses, which have unstable connections to the Internet. Direct Peer-to-Peer sockets can be discarded, as they often have difficulties to handle connection failures. Publish&Subscribe architectures use an external broker, that can keep faulty messages in buffer while needed, and handle complex broadcasts from the server. Section 4.4.1 describes how the MQ Telemetry Transport (MQTT) protocol verifies all these constraints, and how we use it for reliable communications in UbiSMART V3.

A final question about the connection to the gateway is the maintenance of the gateway. Developers must be able to upload remotely a software update, to be notified when a failure happens, and to debug this failure remotely. These challenges are even more complex if gateways use 3G connections, or an external router from the Internet Service Provider, they will hardly be accessible directly through a public IP address. We have decided of two complementary solutions for these questions: making the gateway accessible remotely through a system of reverse proxying, and monitoring the gateways through an external monitoring tool. These are discussed in Sections 4.4.1 and 4.4.3.

4.3.3 The framework Was Not Practically Designed For The End-Users

Another issue with UbiSMART V2 is that it included no interaction with the end-user. Its only outputs were technical logs.

To make UbiSMART a deployment-ready framework for AAL, it should at least respect the following principles:

– Be accessible by the end-user, with no technical knowledge;

– Offer useful services for the end-user;

– Make these services easy to understand, and visually appealing;

– Being secure, so that only authorized users can access the services they want.

A strong work has been done on this matter, so UbiSMART V3 can be more user-centric. Extensive details are given in Section 4.4.2.

4.3.4 Scaling Up UbiSMART For Multiple Deployments

UbiSMART V2 was implemented so that one instance of the program would run for one house. This approach worked with a handful of houses, but it is not scalable for larger deployments. Not only it would induce performance issues, but also an additional human cost in terms of maintenance and installation. Every changes in the program would have to be propagated to every instances. Although some parts could still be automated if necessary, that architecture simply can not be carried forward in a large-scale deployment. Therefore, UbiSMART V3 must be designed to handle multiple houses over a single instance of the program. This is covered in Section 4.4.2, where I describe how we implemented UbiSMART V3 as a web server.

Another feature required by the end-users was to handle the particular case where one user (e.g., caregiver) monitors several houses. The model must therefore integrate multiple houses and users into a single instance of the program, and define relationships between the two entities. The concepts of users and roles is introduced in Section 4.4.2.

Another issue with scalability is on the logistic deployment of smart homes, that is preparing efficiently the hardware for a large number of deployments. In Section 4.4.3, I describe the concept of serial provisioning, to automatize the preparation of smart-home gateways.

4.3.5 Summary

In this section, I have listed every problematic we faced when confronting UbiSMART to the constraint of real deployment. I have given hints about the solutions we have introduced, and I refer each time to the matching paragraph in the following section, for a detailed explanation of our solution.

The major features to be implemented into UbiSMART are as follows:

- A smart home is made of sensors and actuators, plus a gateway (that will handle the processing, or forward to a server);
- Assuming that the processing takes place on a server:
  - Obtain an Internet connection, even in places that don't provide Wifi or Ethernet access;
  - Ensure a reliable connection with the server;
- Being able to perform maintenance on the smart home, even remotely;
- Make the framework accessible and usable to the end-user;
- Scale up the framework for a large number of houses and users.

4.4 Contributions

This section is complementary to the previous Section 4.3. It brings precise answers to every of the problematics listed above.

These answers are the results of a team work, not of an individual work. The features related to the gateway and the smart-home are mostly the contribution of Aloulou and myself. The features linked to the web-server have been managed by Tiberghien and we have been several persons to contribute on it. Although some of these features do not relate directly to my work, I find it important to relate them here, to provide to the reader a complete understanding of our framework for AAL, rather than a partial view. To maintain a fully honest approach, I mention explicitly through this section when a feature has been developed solely by others than me.

4.4.1 Setting Up A Smart-Home

Gateways

Gateways are at the core of smart homes. In UbiSMART V3, gateways forward sensor events to the server. They are also in charge of communicating messages from the server to the smart home, and triggering services and alerts locally.

We use a Raspberry Pi [2] as a gateway (Figure 4.3). It consists of a 35€ computer, the size of a credit-card, and running on a 5V USB charger. It is powerful enough to run a Linux system with reasonable processes, and offers every necessary connections, including USB and Ethernet plugs. Another advantage is that its system is installed through a

[2] `https://www.raspberrypi.org/`

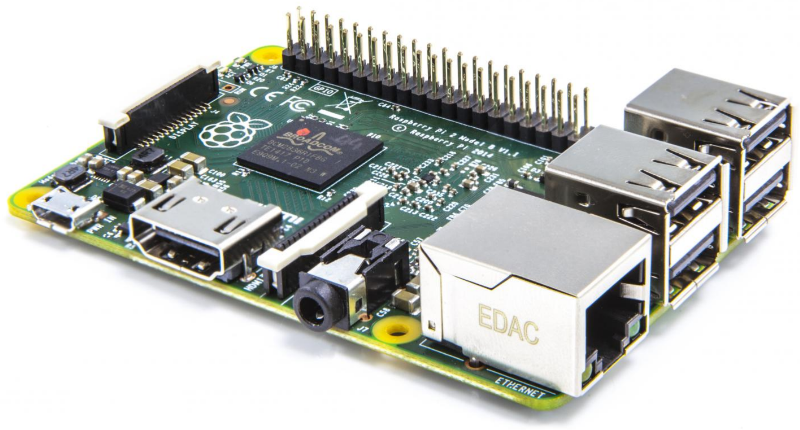


Figure 4.3: A Raspberry Pi — The UbiSMART gateways

MicroSD memory card. Therefore to re-format it, it only needs to replace the memory card.

Originally we had been using BeagleBone Black [3], a competitor of Raspberry Pi, designed to be slightly more powerful. But we have found the Raspberry Pi to be powerful enough, with a larger and more active community of users, and a more reliable product distribution.

Defining a right set of sensors

Keeping in mind that we want our deployment to be as simple and non-invasive as possible, and to keep a low budget to extend more easily, we prefer to use sensors available in the market. We use wireless technologies, since cables are usually not accepted by users, for good reasons.

We have chosen to rely on the X10-RF technology [4], as it has been the most widely used on the last 20 years. By plugging a receiver to our gateway, we are able to receive sensor events, which are then forwarded to the server. The advantage of X10-RF is also its weakness: by keeping a low coupling between sensor and receiver, a X10 sensor is particularly simple to install or replace, but the protocol also lacks some security. The Z-Wave protocol [5] could have been an equally good choice, and it balances X10 strength and weakness through a one-to-one pairing between sensor and receiver.

Our most basic sensors are motion sensors and door sensors (Figure 4.4). In a house with multiple rooms and doors, these sensors already provide rich data, by placing one motion sensor in each room, and a door sensor in the main door as well as on the fridge and oven doors.

However these simple sensors are too sparse in a nursing home, where there are only two rooms, and the main door often remains open. In that case, my colleagues have

[3]`http://beagleboard.org/black`
[4]`http://www.x10.com`
[5]`http://www.z-wave.com`

(a) A motion sensor **(b)** A door sensor

Figure 4.4: Simple sensors using X10-RF technology

prototyped a bed sensor (Figure 4.5) — to know whether a person sits or lies on a bed —, which we have installed in a room in nursing home. The bed sensor consists of a set of pressure sensors, connected to a micro-processor and a ZigBee wireless module. The sensor signals are sent wirelessly to the gateway through ZigBee. The results are interesting in terms of data, but due to a lack of robustness and autonomy, some more work has to be done on it before further deployments.

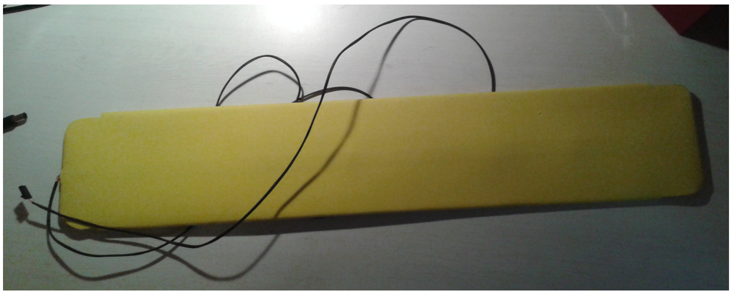


Figure 4.5: A prototype of bed sensor

Using 3G for Internet connection

In houses where there is no stable Internet connection, we need to rely on 3G technology, as usually found on mobile phones. It is possible to connect a computer (in this case a Raspberry Pi) to 3G thanks to USB 3G modems (Figure 4.6), which accept a SIM Card. With the use of the UTMSkeeper software [6], we manage to have a reliable Internet

[6] `http://mintakaconciencia.net/squares/umtskeeper/`

Figure 4.6: A USB 3G modem

connection on our gateways through 3G.

However, this solution is not perfect, and 3G is not fully reliable. Due to the policy of most Internet Service Provider, 3G disconnects every 12 hours, with sometimes difficulties to reconnect. To ensure that the connections is not completely lost, we have installed a watchdog to check the connection and attempt to reconnect. In the worst situation, the gateway will reboot by itself, to restart the connection.

Communications With The Server

The main purpose of the gateway is to act as a bridge between the server and the sensors. As the Internet IP protocol is too heavy, and waiting for the development of 6LoWPAN, it is currently impossible to connect wireless sensors directly to the Internet using Wifi. Instead, wireless sensors implement short-distance protocols, and a receiver is connected to the gateway. Sensor events are parsed by the gateway, which forwards these events to the server.

Using the MQTT Protocol

I have stated in section 4.3.2 that we need a communication protocol that would be light, secure, bi-directional, persistent, and that would assure a quality of service even with unreliable network. I have argued over a Publish&Subscribe model rather than peer-to-peer, to handle complex broadcasts from the server, and to rely on a centralized broker.

Several protocols have been considered. HTTPS can be excluded as it is too heavy, stateless, and single-directional. Simple sockets can be excluded as they are peer-to-peer and they don't ensure a good quality of service. We have tested the XMPP protocol [7],

[7] `http://zeromq.org/`

originally designed for human-to-human communication; it works well, but it is heavy, and not well adapted for machine-to-machine communications A good alternative could have been ZeroMQ [8], an advanced socket implementation, that can work on Publish&Subscribe as well.

But our choice has been the MQTT protocol [9], designed by IBM. It provides a lightweight interface for messages through Publish&Subscribe, and ensures three levels of quality of service. At the intermediate quality of service, a broker ensures that the message has been received by every subscribers, and will retain the message in memory in case it did not. Moreover, the use of MQTT is straightforward, and its implementations handle natively many low-level use-cases such as callback methods for incoming messages, asynchronous publishing, etc.

We use two implementations of MQTT: Paho [10], implemented in Python, and Mosca [11], implemented in NodeJS. We also use Mosca as our broker. During development, we use Mosquitto [12] as an alternative broker, easier to interact with.

The Ubigate Software

Depending on the sensing technology, the Raspberry Pi gateway reads sensor events from various softwares. For X-10 RF, there exists a software named Mochad [13], to receive events from the X-10 receiver. Z-Wave and Zigbee each have their own software tool to interact with sensors.

I have developed the Ubigate software on the gateway-side, in order to aggregate sensor events from various sources, and facilitate the use of MQTT. It is developed in Python, and mostly consists of a wrapper around the MQTT Paho library. In the background, Ubigate will always ensure that the MQTT connection is established (and reconnect if necessary). If a message is sent through Ubigate, it will be saved into a persistent buffer until the message has been received and acknowledged by the server. This way, even in case of connection failure, we can ensure that no message will be lost.

Ubigate uses a plugin-based architecture, so that tools that need to communicate with the server are implemented as Ubigate plugins. The most actively developed plugin for Ubigate is named *marmitek-gw*, and retrieves X10 FR events from the Mochad software. At the moment, Ubigate only support sensor plugins, but we may consider more advanced use cases in the future, such as interactions with a smartTV based on the same model, through Ubigate.

[8] `http://xmpp.org/`
[9] `http://mqtt.org/`
[10] `https://pypi.python.org/pypi/paho-mqtt/`
[11] `http://www.mosca.io/`
[12] `http://mosquitto.org/`
[13] `http://sourceforge.net/projects/mochad/`

Maintenance

Smart homes are sensitive to failures, whether from a lost Internet connection, the user accidentally unplugging the framework, a broken sensor, or a faulty software. I have observed these failures multiple times from our deployments.

To handle these failures, we have made our gateway as robust as possible. If a faulty software stops, it will be restarted automatically. If the gateway happens to reboot, every necessary softwares will be started properly. Sometimes the gateway even reboots by itself, in case the Internet connection is lost, to properly reset the connection.

Yet, a human maintenance is sometimes necessary. Additionally, with an iterative development cycle, we frequently need to upgrade the software stack on the remote gateway. Since it is costly in time and money to physically send a person in the house for maintenance, we have worked on remote maintenance. The gateways are not be accessible on the Internet out-of-the-box, as they are hidden behind a router or the 3G provider.

Therefore, I have setup a reverse proxy. In a reverse proxy, the gateway establishes a persistent connection to a publicly available server, on a specific port. From this point, every packet sent to the public server on this port are forwarded to the gateway (Figure 4.7). This way, the gateway is publicly accessible from the Internet, and a client can seamlessly connect to it from anywhere. We use the secure protocol SSH to perform the reverse proxying.

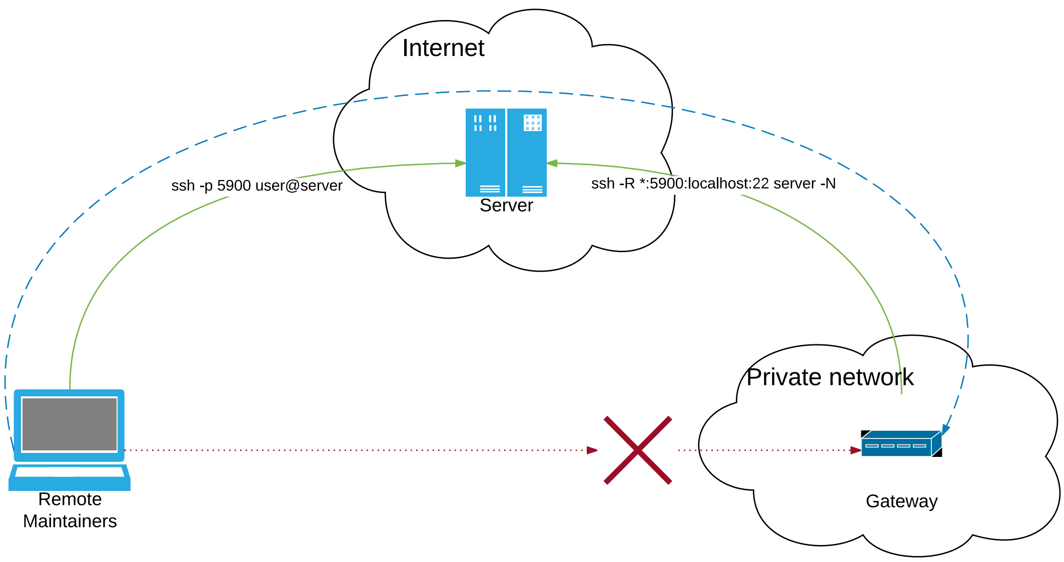


Figure 4.7: The concept of reverse proxying

4.4.2 Making UbiSMART More User-Centric

Another issue we had been facing with UbiSMART V2, is that its interaction with the user were limited to sensor events and alerts. In UbiSMART V3, we have decided to improve this point in particular. Users should be involved in the deployment; and to do so, they should benefit from the framework.

Redesigning The framework As A Web Server

We have decided to redesign UbiSMART as a web-server. This work has mostly been carried by Tiberghien. Although this approach is rather unusual in the literature, we see several advantages in turning UbiSMART into a web-server:

- It can be accessed from anywhere, with no need for the user to install anything;
- Web-server are powerful to handle intensive input/output requests, with CPU-intense processing in the background:
  - This use-case can be transposed to a smart-home;
  - Input and output requests do not have to be in HTTP, and modern web frameworks can process other protocols, such as MQTT;
- HTML/CSS/Javascript offer a unified and standardized language to design user interfaces;
- Web frameworks already provide high-level methods to manage user sessions, saving models in database, etc;
- Web development is supported by an extremely large and active community;
- With the recent development of the Mobile Web, and tools like PhoneGap [14], it can easily be adapted for smartphone and tablets.

In terms of tools, my first approach was to use Python's Flask microframework [15]. Indeed, Flask provides every required methods to simply create a web server, but remains unobstrusive on the backend, leaving flexibility for complex and unusual applications like UbiSMART.

But with further investigations, we have decided to go for NodeJS [16] instead, as it has a more promising future, and it offers better performance. Indeed, where most web servers simply parallelize requests with threads, NodeJS uses asynchronous methods, so that it can handle a large number of requests with a single instance and a single thread. Multi-threaded models can be effective with CPU-intensive processes, but in the case of a web-server, the bottleneck is more often in accessing and serving data, and asynchronous development is shown to perform better.

On top of NodeJS, we use the SailsJS framework [17]. SailsJS is a reimplementation of

[14] `http://phonegap.com/`
[15] `http://flask.pocoo.org/`
[16] `http://nodejs.org`
[17] `http://sailsjs.org/`

the famous Ruby On Rails framework [18], using NodeJS rather than Ruby. It is not as mature as Ruby On Rails, but already offers better performances thanks to NodeJS. It uses the REST architecture [310] and the Create, Read, Update, Delete (CRUD) model [311] as much as possible, following the *Convention over Configuration* paradigm. *Convention over Configuration* means that the default behaviour of the framework, — when the developer follows its convention —, will be particularly efficient and convenient; nevertheless, it is always possible to bypass it with additional configurations; therefore it is never restrictive.

I have not being involved directly in the original design of UbiSMART using SailsJS, as this work has been carried by Tiberghien and Bellmunt. However, I have participated in several of the services described below.

Secured Access For The Users

We have identified several types of users: *Residents*, *Caregivers*, *Technicians* and *Doctors*. Each users can be supervising several houses, but cannot access houses they are not attached to. Therefore we have created a model of user into the UbiSMART web server. Users can log in with their credentials (Figure 4.8), and will access only what they are authorized to see (Figure 4.9).

It is also important to notice that the server is secured with a SSL certification.

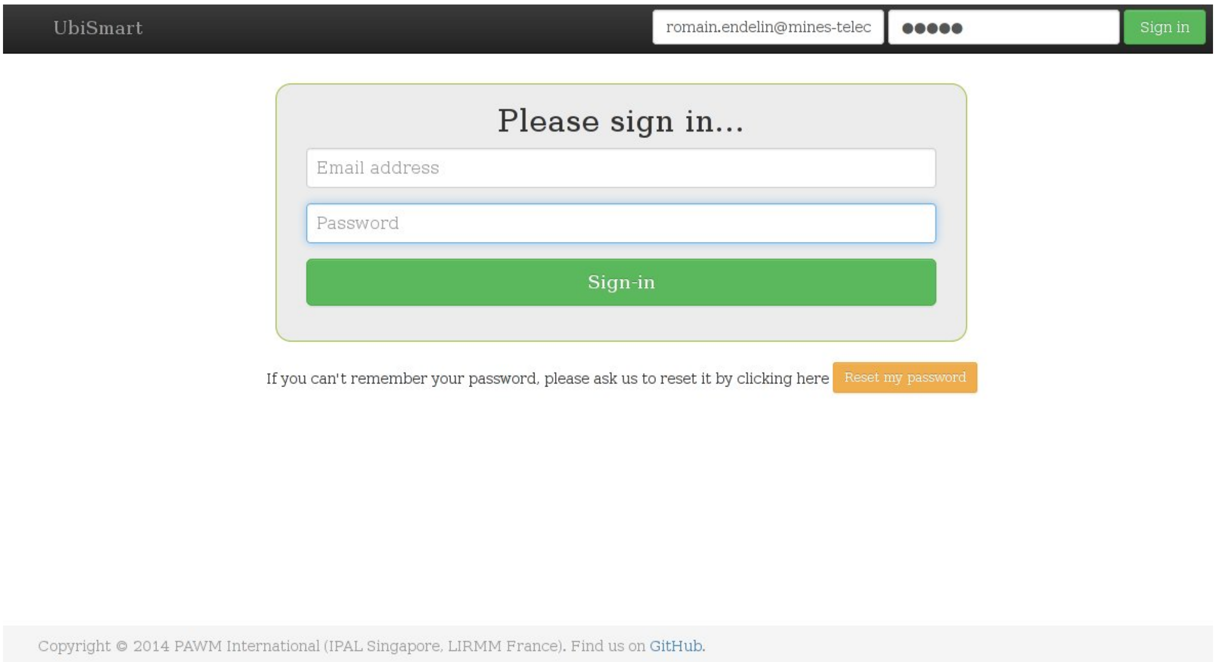


Figure 4.8: UbiSMART Login Interface

[18] `http://rubyonrails.org/`

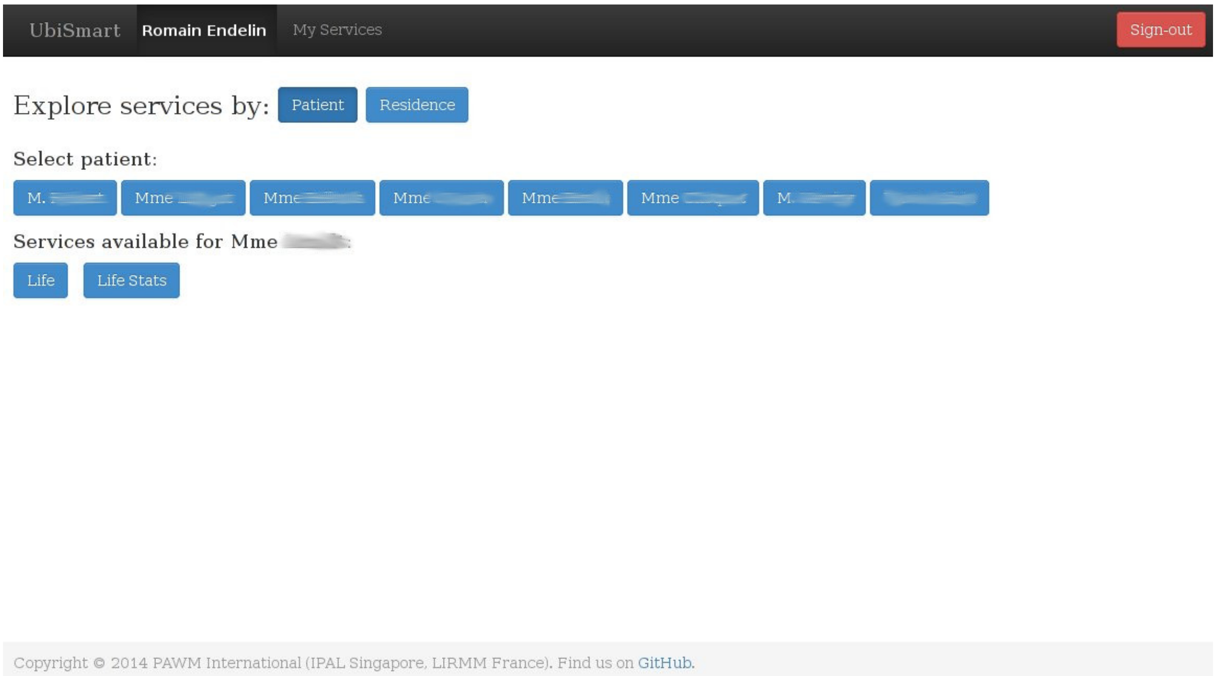


Figure 4.9: UbiSMART House Menu

User-Friendly Services

We have developed several visualisations and other services for the users. This way, users can see what the resident has been doing over the last 10 minutes, 1 hour, 3 hours, 6 hours, 1 day or 1 week (Figure 4.10). Nurses can also have a complete view of a corridor inside the nursing home, so they can quickly identify falls (Figure 4.11). Similarly, we have created a view of individual houses, to visualize the resident's activity (Figure 4.12). Technicians will also be interested to visually observe the sensor events (Figure 4.13). For doctors, we have developed long-term statistics visualisations, for sleeping cycles and uses of toilets (Figure 4.14).

These visualisations are created using the D3.js library [19].

Additionally, we have implemented alerts through smartphone applications (Figure 4.15), SmartTV (Figure 4.16) and email. However, these alert remain in early stage, and further work is ongoing to extend them.

4.4.3 Seamlessly Extending To Multiple Deployments

Another limitation of UbiSMART V2 was that it was designed for only one user.

With UbiSMART V3, we have taken every required actions to seamlessly extend for multiple users in multiple deployments.

[19] `http://d3js.org/`

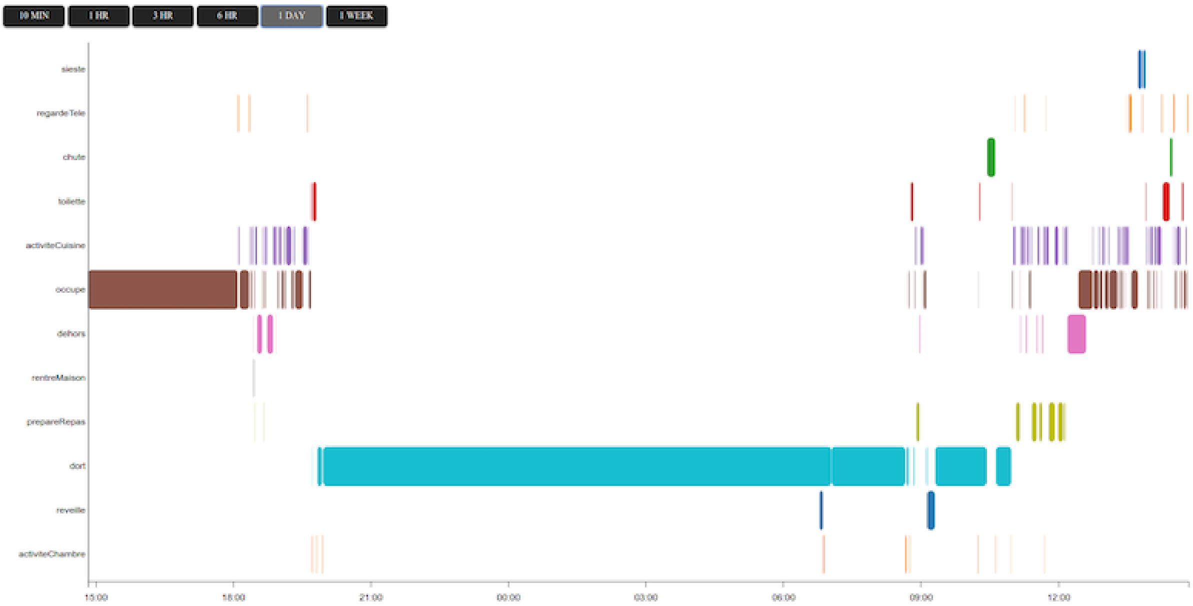


Figure 4.10: Visualizing Activities in UbiSMART

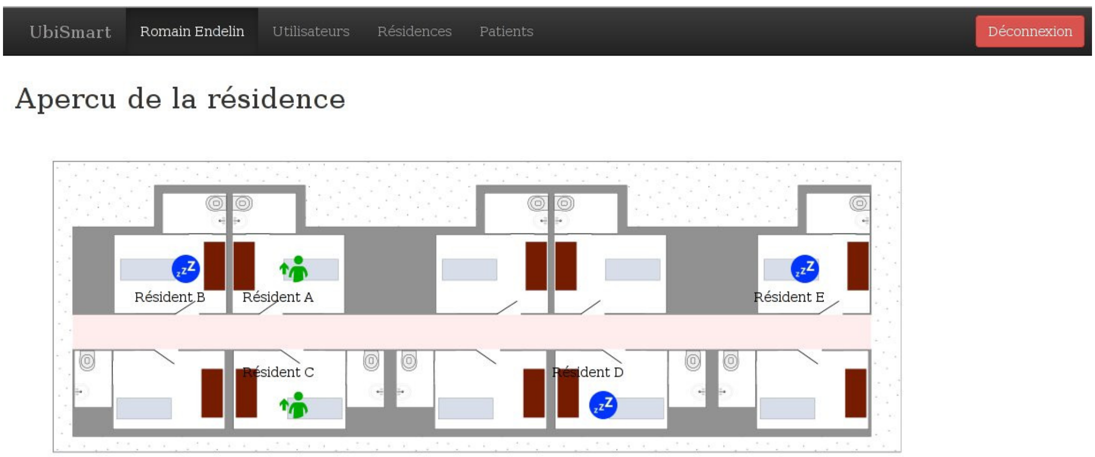


Figure 4.11: Visualizing a Nursing Home Corridor in UbiSMART

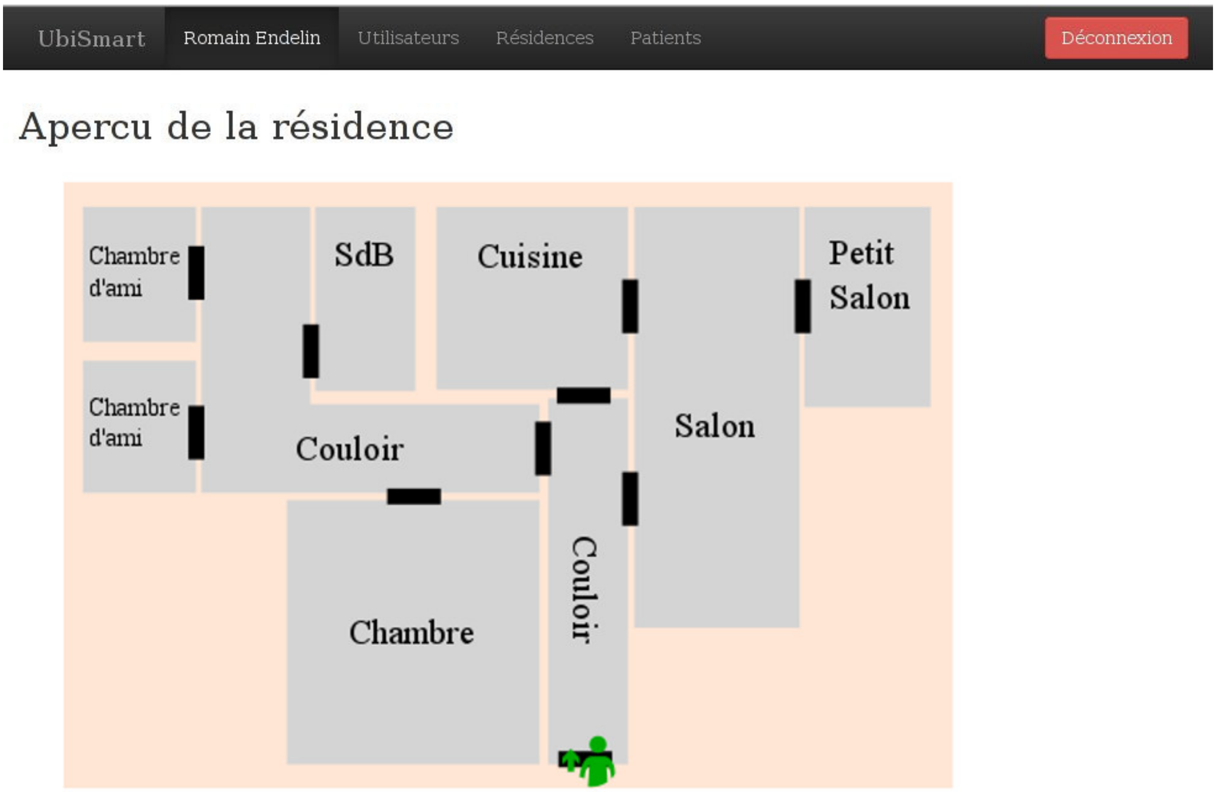


Figure 4.12: Visualizing House in UbiSMART

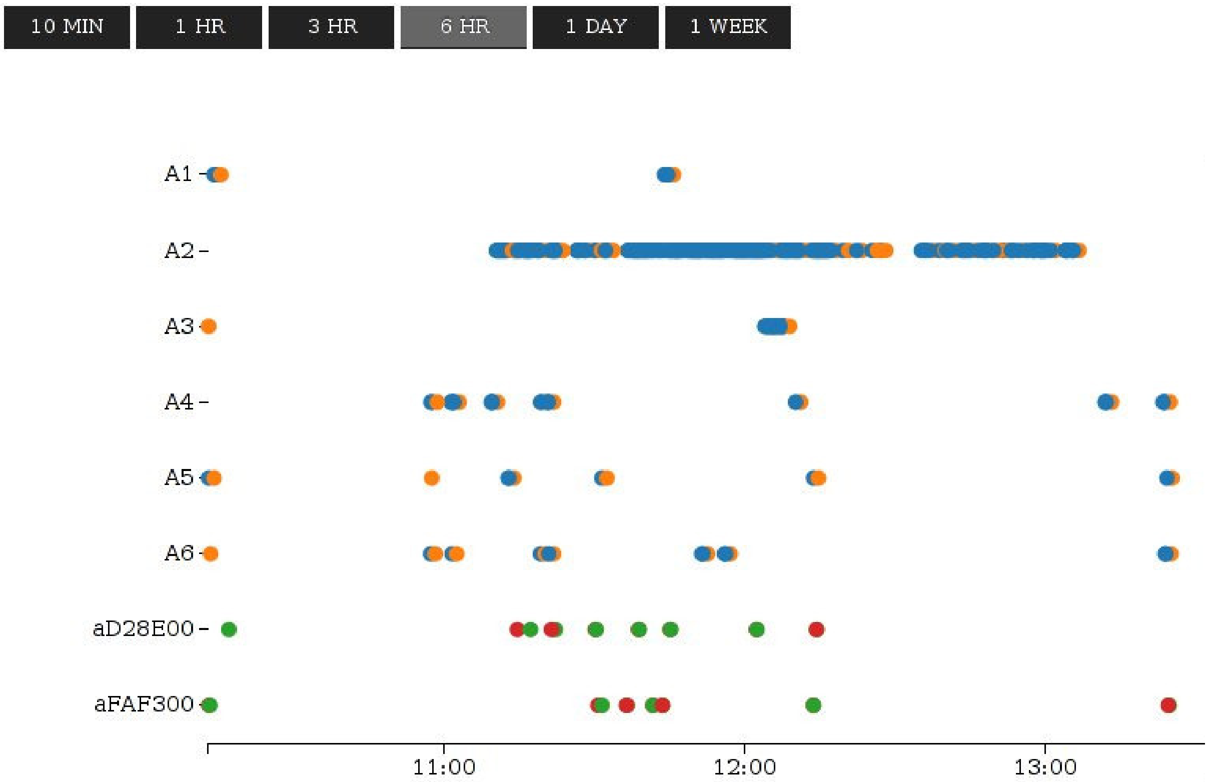
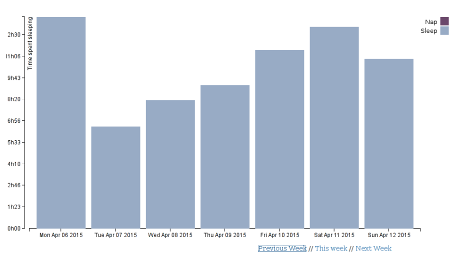
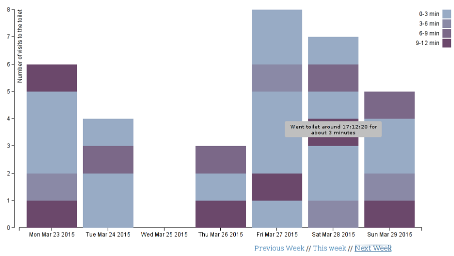


Figure 4.13: Visualizing Sensor Events in UbiSMART



(a) Statistics on sleep



(b) Statistics on toilet use

Figure 4.14: Statistical visualisations in UbiSMART

Figure 4.15: UbiSMART Alert on Smartphone

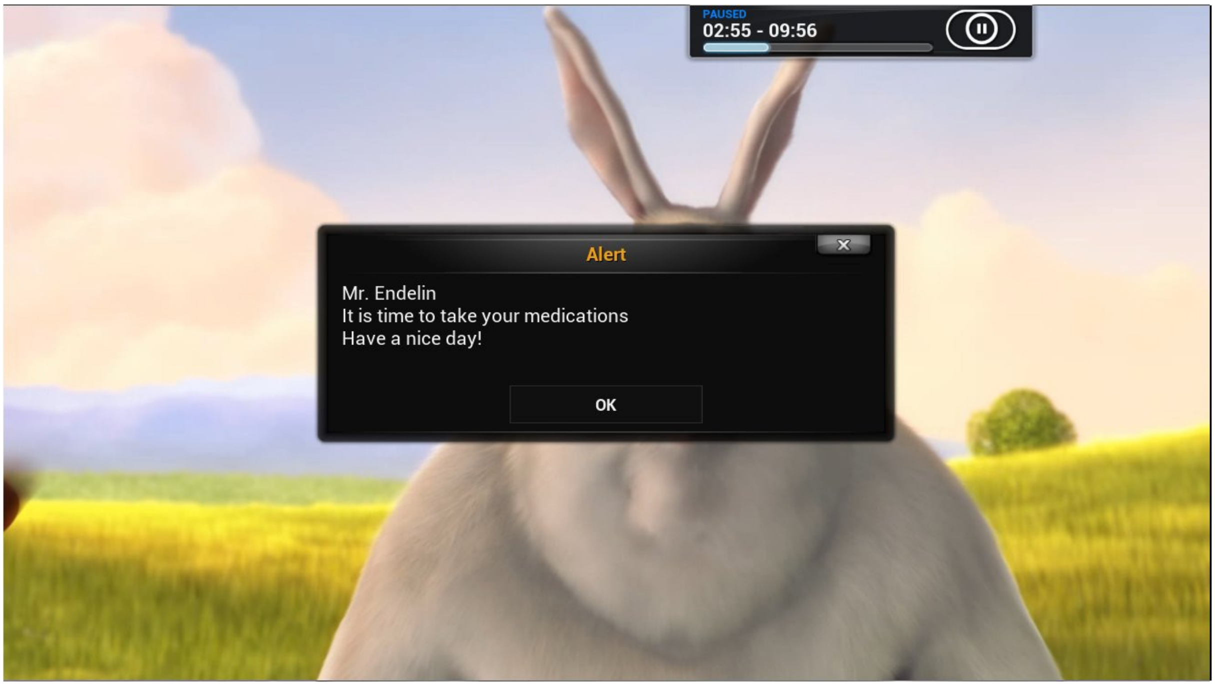


Figure 4.16: UbiSMART Alert on SmartTV

Gateways

Each smart-home includes a gateway, so it needs to be duplicated for every new deployments. Considering a large-scale deployment, we need to make the payload of creating a new house as little as possible.

Serial provisioning and maintainance

First of all, we have created a process to automate the installation of gateways. Gateways essentially consist of a Linux distribution, with some software installed in it, and some specific configuration. We have defined all these specificities in a declarative format, using Ansible [20]. Ansible is a simple and powerful automation tool. Its major benefit is that Ansible scripts (named playbook) are idempotent, so that executing it one thousand times will have the same effect as execute it one time. Through idempotency, Ansible can manage both the installation of the gateway (we name it provisioning), and the further system upgrades. Moreover, Ansible manages a list of gateways with their IP addresses, so that the developer can upgrade every gateways at once from Ansible.

Thanks to Ansible, preparing a new gateway only require four actions: copying a Linux image to a microSD card, plug it to the Raspberry Pi gateway, turn on the gateway, and execute Ansible for the new gateway. After new features are implemented, upgrading the gateways then simply requires to execute Ansible.

The bottleneck of this approach is that the initial provisioning may take several minutes. Aloulou has then put a shortcut in place. Rather than executing the whole script

[20] `http://www.ansible.com/home`

for every gateways, an Image file (.iso) has been created, including a barebone linux, with every useful softwares already installed. For a new gateway, the technician repeats the same steps as usual: copy the Image file to a MicroSD card, plug it to the Raspberry Pi, turn on the gateway, and execute Ansible. But this time, the ansible script will take only a few seconds to execute, as it will only update the configuration, and the softwares are already installed.

An interesting extension would be to wrap the entire process inside Vagrant, so that the setup could take place on the virtual machine, whose image would then be copied to the MicroSD card. This way, the setup process would not even require a Raspberry Pi.

Automatic Configuration of the Gateway

Each house is different, and the gateways need to maintain a configuration file, where they describe sensors and devices installed on this house, as well as other informations.

To facilitate this process, we now configure the house from the UbiSMART server, using a web interface. The configuration is then sent to the gateways through MQTT. When receiving this message, instances of Ubigate will save the new configuration.

The process goes even further, with sensor discovery. If an event is received from an unknown sensor, the gateway will forward it to the UbiSMART server, asking the administrator to identify or reject this new sensor.

Monitoring

Monitoring and maintaining gateways is a heavy tasks. It involves for the administrator to manually connect to the UbiSMART web application everyday, checking that sensor events are correctly received in each house. If they are not, the administrator must try to manually connect to the gateways, looking for failures. This process is critical, but cannot scale up to a large number of deployment.

To overcome it, we have setup a monitoring framework, using Zabbix [21] (Figure 4.17). The framework will check the following vital signs from the gateways:

- Is the gateway reachable;
- Uptime of the gateway;
- Are every required processes running;
- Do we observe any error in the log file;
- CPU/RAM load;
- Are every devices plugged to the gateway;
- Is the clock on the right time;
- Is network behaving correctly;

[21] http://www.zabbix.com/

– Do we receive every sensor data correctly.

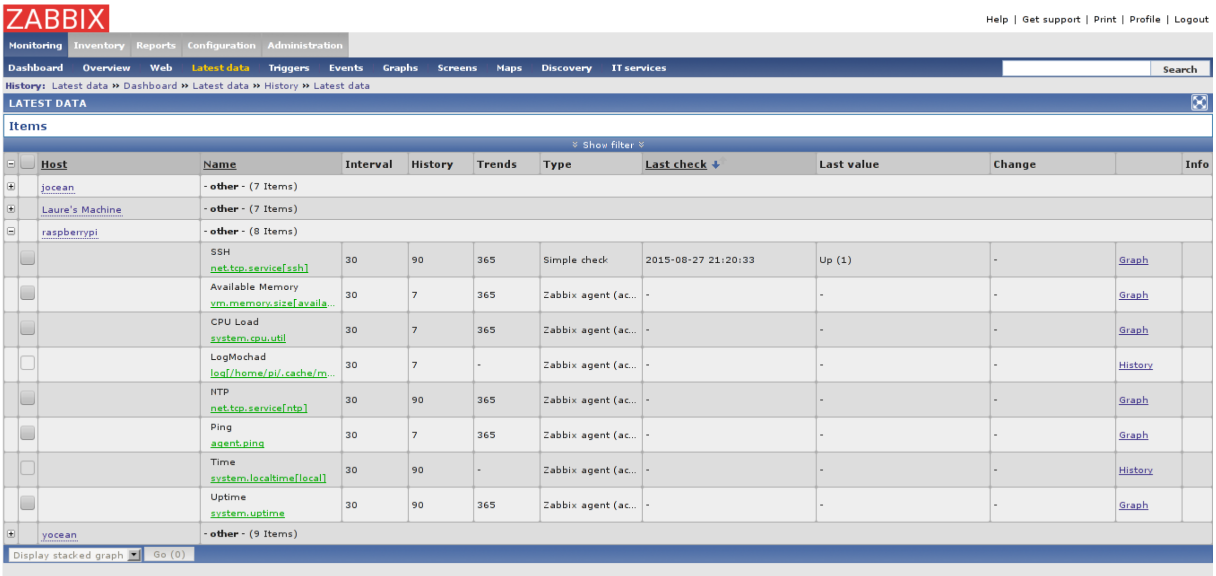


Figure 4.17: Our gateway monitoring interface

Additionally, the monitoring framework will also check the availability and health of the UbiSMART Server.

In case a problem is noticed, several procedures have been put in place by the monitoring framework to repair it automatically, if possible. This involves restarting services that are not running, or resetting the clock. For more complicated cases, email alerts will be sent to the administrator.

This work has been implemented by Tiberghien, with the help of Bentes and Crémieux.

Program Execution

Our server also has to scale up for multiple deployments. With UbiSMART V3, only one instance is running and manages every houses, whereas in UbiSMART V2 each house required one instance.

Performances

In terms of performance, we benefit from the NodeJs language, that is particularly efficient to handle multiple requests at once. Therefore, several gateways can transfer sensor events, at a pace of one event every five seconds, and multiple users can access the web interface, without causing overload. Figure 4.18 shows a load test we have performed on the UbiSMART V3 performances [50].

UbiSMART V3 is still facing some performance issues in case too many data are saved in the database, forcing us to regularly make a backup of the database and empty it. These issues should be addressed in a near future.

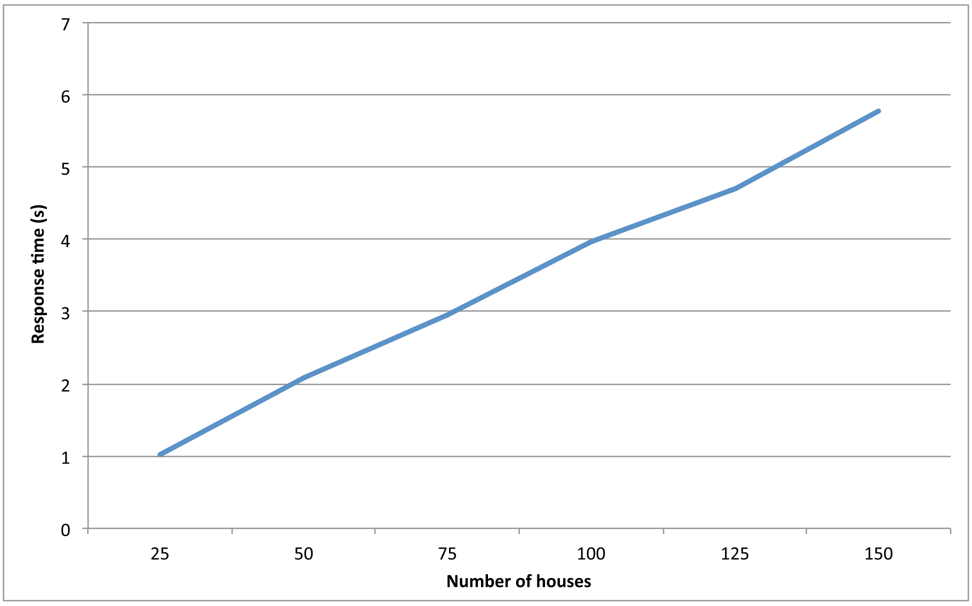


Figure 4.18: Load test of UbiSMART V3

Modeling Multiple House

In terms of model, UbiSMART V3 introduces the concept of neighbourhood. A neighbourhood is composed of multiple houses. It is even possible to have a neighbourhood composed of smaller neighbourhoods, like a city composed of buildings, composed of apartments. The main interest of neighbourhoods is a fine-grained tuning of users' privileges. For instance, a caregiver can be attached to an entire residence. Another interest of neighbourhoods is that we can attach a service to a neighbourhood, such as a view of a corridor inside a nursing home.

We have also seen in Section 4.4.2 that users can now securely access to the houses and neighbourhoods they are attached to. This is called *User Interface Plasticity*, and it is currently been researched more deeply in the ongoing thesis of Bellmunt.

Reasoning

Another subject that has to be taken into account when scaling up is the reasoning. In UbiSMART V2, the reasoning was manual. This means that every new deployment required the administrator to manually copy a template of ontology and edit it, to describe the rooms and sensors. The same process had to be repeated when adding / removing sensors to a deployment.

In UbiSMART V3, we have introduced a Semantic Plug&Play, originally imagined by Aloulou during his thesis. With this concept, it is possible to update the description of a house from the web interface (Figure 4.19), and the ontology will automatically be updated accordingly.

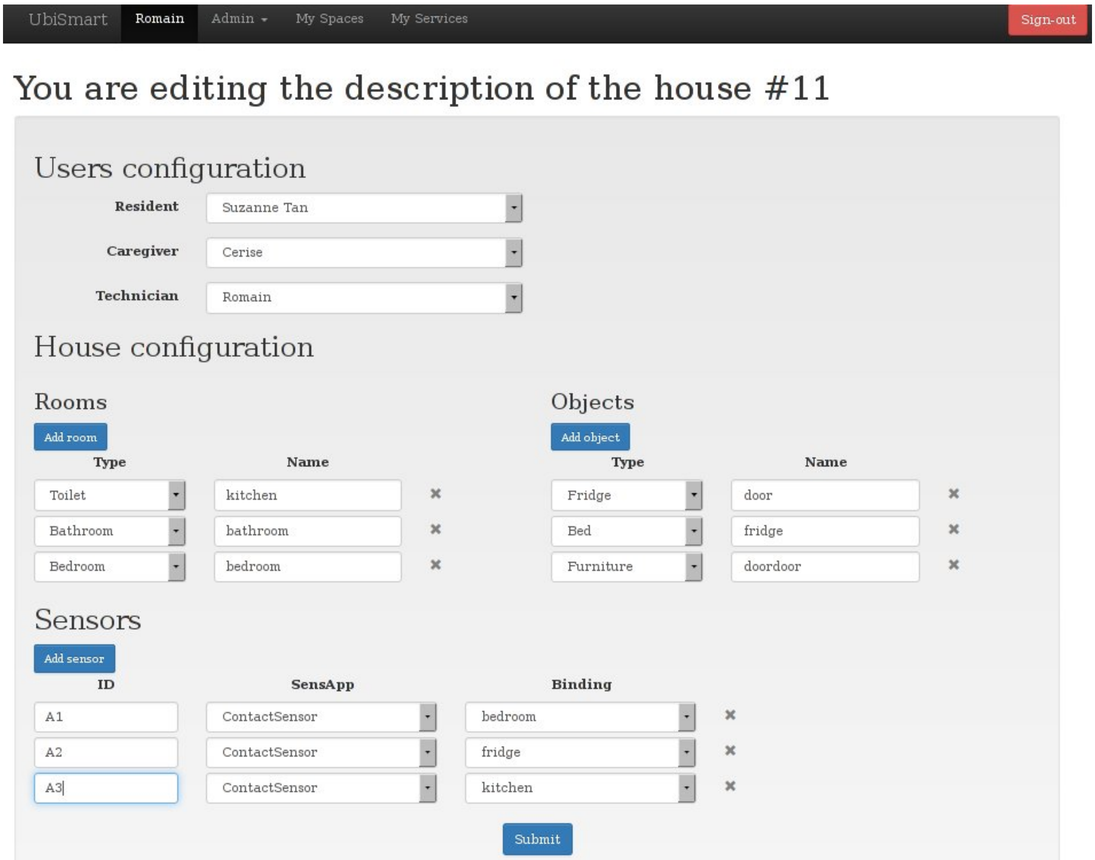


Figure 4.19: Describing a house through a web interface

4.5 Discussion

4.5.1 Outcomes

Figure 4.20 shows the current features of UbiSMART V3, as described in this chapter. Following, I discuss the feedback we have obtained from our current deployment, and the lessons we have learned.

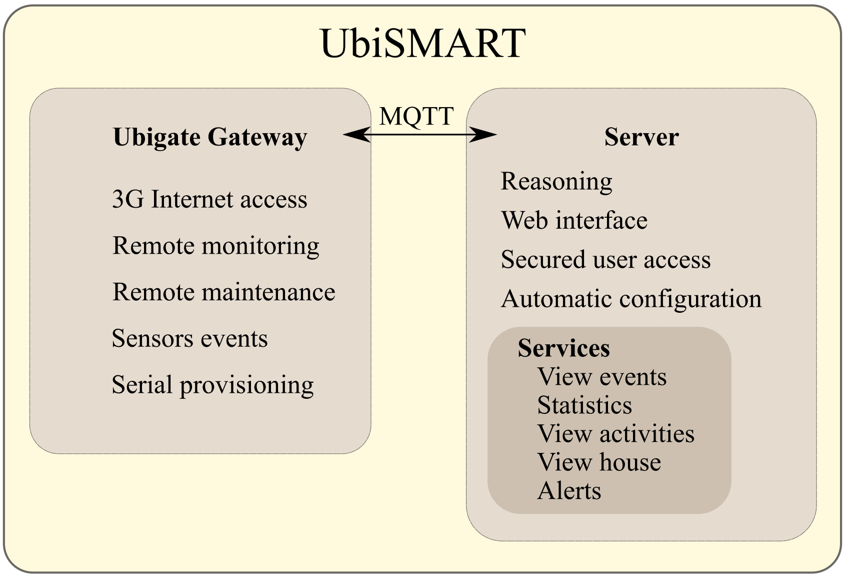


Figure 4.20: The different modules in place in UbiSMART V3

Feedback From Our Current Deployment

During our recent deployment, we have observed several requirements that were not originally verified, and we have worked on each of these.

First, our current framework, UbiSMART V3, is particularly easy to setup. By running a script, and doing some configuration through the web interface, we are now able to get a house ready to deploy. Once the gateway and sensors are ready, we have deployed houses in less than 20 minutes, with no technical skill involved. In the past, such deployment would have required a day, and involved to setup the framework locally. Our system also takes really little space for the resident. The gateway is only the size of a matchbox, and the sensors are simply attached to the wall with blu-tac, easy to remove at any time. As a final product for AAL, that would be ready for large-scale deployment, we hope to provide a “smart-home in a box”.

We have made a particular effort to make the system reliable, even with a sometimes non-reliable Internet connection over 3G. Moreover, maintenance and debugging can be performed remotely.

Our UbiSMART V3 web interface is currently being used by nurses in the deployed nursing home, as they can instantly visualize what happens in the rooms of the residents.

On a more technical side, we also have obtained 462,284 sensor events and inferred 84,351 activities over a period of 386 days. These data will be leveraged to improve the reasoning, as described in Chapter 7 of this dissertation, and validated in Chapter 9.

Lessons Learned

The first lesson I have learned from our deployment is that there is actually a large demand for AAL systems. Nursing homes are particularly interested on it, as it can optimize the activity of the caregivers, and make some long-term evolutions appear on the residents. There is also a demand from families, expecting to be reassured about their elders, and to react faster in case of an accident. As I have shown in this chapter, the technological barriers are being opened, and I claim that there is no more technical deadlock for a large-scale deployment (although we are still facing a barrier on reliable activity recognition). One may expect a large development of industrial AAL solutions before long.

Another lesson I have learned is the importance of involving the users in the process. Not only it benefits to these users, as they can use our services and they feel more respected; but it also benefits to the researchers, as it helps us to define our needs better, and adapt our work from their feedback. Often times we have observed a gap between laboratory discussions — that tend to be particularly technical and to bring up complicated use cases — and discussions with the end-users, that usually highlight simple use-cases from common sense, fitting much better with their daily uses.

I also insist on the fact that development tasks cannot be neglected, side-by-side with research. These are complementary, and in applied research like AAL, one should keep a balance between theoretical and technical work.

One last observation I have made is that some technical difficulties will never appear in laboratory settings. Yet these difficulties have to be taken seriously, and should not be postponed. Laboratory settings are merely an attempts to reproduce real situations with more flexibility, and should never be taken as the norm in AAL. If a framework works in laboratory, but fails in a real situation, we must simply consider that if fails.

From all these observations, I can only acknowledge how much benefits we had from deploying our system in real situations.

4.5.2 Openings

In this section I have presented many features that we have implemented into UbiSMART. Thanks to these features, UbiSMART is now functional and running full-time in our deployment.

Yet, there are some improvements that can be made, and I will introduce some of these in this section.

Ubigate As A Distribution

Currently the gateway is a installed through an automatic process, and communication with the server is performed through a software we have named Ubigate. I would like

Ubigate to become a complete distribution, in the Linux definition of distribution. Ubigate would have a disk image to install it. It would be built upon Linux, and contain a set of softwares for Internet communication, communication with sensors, communication with the server, service provisioning, monitoring, and any other software that would be necessary. These softwares would be installed and maintained though a package manager.

These concepts of package manager, installation process, etc. can of course be generalized, and in a sense the current version of our gateway already verifies these criteria. Yet I would like the concepts to be more formalized. In particular, the Ubigate Distribution should rely on a powerful life-cycle, with versions and releases.

Develop More Services

The current version of UbiSMART already provides services, whether informative visualisations that users can access on-demand, or notifications in case of alerts, through email or smart-phone application. Yet, some improvement can be done in terms of output devices.

First, some more devices could be integrated, such as SmartTV, speakers, mobile phones with SMS notifications, and an improved email service. Additional visualization services could also be useful to observe long-term behaviour change, through statistical views.

More generally, the service provisioning should be made smarter, by adapting to the user. This is named User Interface Plasticity, and Bellmunt is currently focusing on it for his upcoming thesis. The goal would be to provide the right information to the right person at the right time, displayed the right way, using the right device.

Reworking the User Experience

The web interface of UbiSMART — as shown for example in Figures 4.9 and 4.10 from Section 4.4.2 — is functional. But in term of User eXperience (UX), it may probably be improved by the advices of a professional designer, and by gathering feedback from the end-users themselves. This is particularly important, first because we aim at making our development user-centric, but also because many of our end-users are aged of sixty years-old or more, and may not be familiar with the technologies. Thus a clear and intuitive interface could greatly benefit to the framework.

Pre-Processing Of Sensor Events

Some sensors, like accelerometers, produce low-level signals, that needs further processing to infer higher-level events like "shaking" or "falling". It would be interesting to formalize the pre-processing step from within UbiSMART, through modules named *SensApps*, that would be called automatically on new signals. The concept of *SensApps* had been envisioned by Tiberghien during his thesis.

Go Beyond Smart-Homes

An interesting opening for UbiSMART would be to extend it beyond the use-case of smart home. It could become a framework for smart-city, including numbers of smart-homes but also civil services connected to the city. In a sense, this already has been started through the introduction of neighbourhoods. Moreover, UbiSMART V3 has been developed on a modular approach, where the core is agnostic to services and sensing technologies. But additional work should be done before one could really use UbiSMART to manage a Smart-City, mostly in generalizing the concepts of house, neighbourhood and users.

To go even further, UbiSMART might go beyond AmI, and be as global as a framework for the IoT. This is not possible with our current design, as activity recognition is the keystone of our application, and it is too specific to AmI. For instance, a smart garden, or a smart agenda are interesting use-cases in IoT. They require a certain level of reasoning, but this reasoning would appear to be very different from the activity recognition performed in AAL. Thus, a large amount of work would be required to generalize even the very concept of reasoning, and such work is unlikely to be undertaken in a near future.

4.5.3 Limitations: Activity Recognition

Through this chapter I have described all the work done to make our AAL framework — UbiSMART — ready for real large-scale deployment. In my humble opinion, UbiSMART is a promising application, and the critical criteria for it to run in deployment are now verified. We have also validated it towards three houses and five rooms in a nursing home, with a positive feedback, both technical and human (Chapter 9).

Nevertheless, there remains one challenge of UbiSMART that has not been addressed in this chapter, and which is a keystone of AAL: the ability to recognize activities reliably. Even the best AAL system can not run properly without a reliable activity recognition engine to support it.

Chapter 5 introduces a rapid prototyping framework for activity recognition, while Chapters 6 and 7 describe two methods to improve the reasoning for activity recognition.

Many programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program.

Linus Torvalds

5 Prototyping Activity Recognition Reasoning Engine

5.1 Motivations

Early in this thesis work, I have relied on the legacy UbiSMART reasoner for activity recognition, developed by Tiberghien and Aloulou. Yet, it suffered from one major flaw. Rules had been written manually, with short-term delivery in mind. Necessary evolutions over time had not been considered, and modifying the reasoning engine appeared to be impractical. I have come to realize the important of continuous improvement for activity recognition, in order to support my thesis work and further developments.

In consequence, I have developed a prototyping framework for activity recognition, named Knot. Knot is used for the following two use-cases:

– Loading AAL dataset, visualize them, and analyze them;

– Rapid prototyping of the reasoning engine.

Disclaimer: *Some of the code listings presented below have been slighly simplified, for readability purpose.*

5.1.1 Features

Knot has been developed around the following features, each of which are described in this chapter:

– A prototyping environment (Section 5.2):

  – Prototyping notebooks;

  – Scientific programming libraries;

  – High-level libraries to interact with RDF;

  – Fully modular code.

– A data analysis library (Section 5.3):

  – Data structures;
  – Data loading and pre-processing;
  – A all-in-one function for data visualization.

– Ability to run the reasoner on-demand (Section 5.4):

  – Modular reasoner;
  – Explicit logger (observes changes in the ontology);
  – Run the reasoner on various data sources.

– Simulating realistic activities (Section 5.5).

Additionally, the following applications have been developed on top of Knot:

– Multiuser detection (Chapter 6);

– Accuracy measurement, based on ground-truth observations (Chapter 7);

– Dynamic creation of houses and rules (Chapter 8).

– Notify inconsistent rules (e.g., missing sensor, conflicting rules) (Chapter 8).

5.1.2 Relations Between Knot And The UbiSMART Reasoner

Although Knot is reasoner-agnostic, its construction has been tightly coupled with the UbiSMART reasoning engine.

During their thesis work, Tiberghien and Aloulou have developed an ontology for smart homes, and a set of rules for Activity Recognition. The approach is to use knowledge-based reasoning, through ontologies and logical rule engines. Knowledge about the environment (e.g., rooms, sensors) are saved into an ontology, based on N3 or RDF languages.

The UbiSMART reasoner had been well-integrated into the UbiSMART framework, and runs in real-time in the deployed houses. With Knot, I have developed an alternative implementation of the reasoner with additional features and an increased flexibility for prototyping. More details on this implementation are given in Sections 8.2 and 5.4.

Introducing The Euler Rule Engine

In the UbiSMART reasoner, a set of logical *IF-THEN* rules is defined into the ontology (Listing 5.1). When a new input enters the ontology (e.g., sensor event), the Euler Rule Engine [34] will automatically trigger the rules with validated conditions, and infer higher-level knowledge (e.g., context, activities) into the ontology. The newly discovered knowledge will itself validate other rules' conditions, and a new iteration of Euler will then infer even higher-level knowledge. The reasoning process stops when a stable state is reached, and no further knowledge is discovered.

```
{
    ?x family:parent ?y.
    ?y family:brother ?z.
} => {
    ?x family:uncle ?z.
}.
```

Listing 5.1: N3 rule, to be inferred by the Euler Rule Engine

Introducing Cerebrations And Cogitations

Tiberghien's Thesis also introduced the concepts of *cerebration* and *cogitation* [31]. A *cogitation* is an iteration of the rule engine over our ontology. However, some operations are particularly hard and inefficient to perform with ontological reasoning (e.g., arithmetic and disjunctions).

To address these difficulties, *cerebrations* are classic imperative algorithms, which obtain their inputs from the ontology, and whose output will be saved into the ontology too. At each iteration of the reasoning, every *cerebrations* are executed sequentially between two iterations of the *cogitation*. An example of cerebration is an algorithm to measure the quantity of movement in a room over a time-window; its input are sensor events saved into the ontology, and the output is saved into the ontology as a property of the room.

5.2 Technical Specifications

Knot has been developed with the Python language. Alternatives options would have been Matlab, R, Octave, Java, C or Julia. Table 5.1 summarizes the features of each of these languages, and justifies the choice of the Python language.

On top of Python, Knot uses the following libraries:

Numpy Scientific programming [1] (described in Section 5.3.2);

Pandas Data structure [2] (described in Section 5.3.2);

[1] www.numpy.org
[2] http://pandas.pydata.org

Language	Scientific tools	Maturity	Versatility	Abstraction	Performance [312]
Python	✓	✓	✓	High-level	Average
R	✓	✓	✗	High-level	Slow
Matlab	✓	✓	✗	Very High-level	Slow
Octave	✓	✗	✗	Very High-level	Slow
Julia	✓	✗	✗	High-level	Good
Java	✗	✓	✓	High-level	Average
C	✗	✓	✓	Low-level	Good

Table 5.1: Comparing programming languages for the development of Knot

Bokeh Data visualization [3] (described in Section 5.3.3);

RDFLib Working with RDF and SPARQL [4] (described in Section 8.2);

Simpy Discrete Event Simulation [5] (described in Section 5.5).

Knot encourages the use of the Jupyter [6] development environment (Figure 5.1), which has also been designed with prototyping in mind. It consists of notebooks, rendered in a web interface. The notebooks contain blocks of code, that can be ran independently from each others, while sharing the execution environment. Through notebooks, the user can interactively test his code, visualize figures, and insert blocks of formatted text. The content of a notebook is also saved for later use, and can easily be shared.

I argue that the use of these tools is an important part of the prototyping process, and must be documented along with the framework's architecture itself.

5.2.1 Architecture

Figure 5.2 describes the file structure of the Knot project. Notebooks and dataset files are saved in their respective folders. The *knot/* folder includes python packages for data analysis (*data_analysis/*), reasoning (*reasoning_engine/*, *sinks/*, *sources/*), houses and rules factory (*model/*) and simulation (*simulation/*). The *presets/* packages includes predefined structures: pre-processing steps for data analysis (*data.py*), house and rule definitions (*house.py*) and reasoning engine configuration (*reasoner.py*). It should be noted that at the moment Knot does not include executable application, and it is meant to be loaded as a library, inside a Jupyter notebook.

[3] `http://bokeh.pydata.org`
[4] `http://rdflib.readthedocs.org`
[5] `https://simpy.readthedocs.org`
[6] `http://jupyter.org`

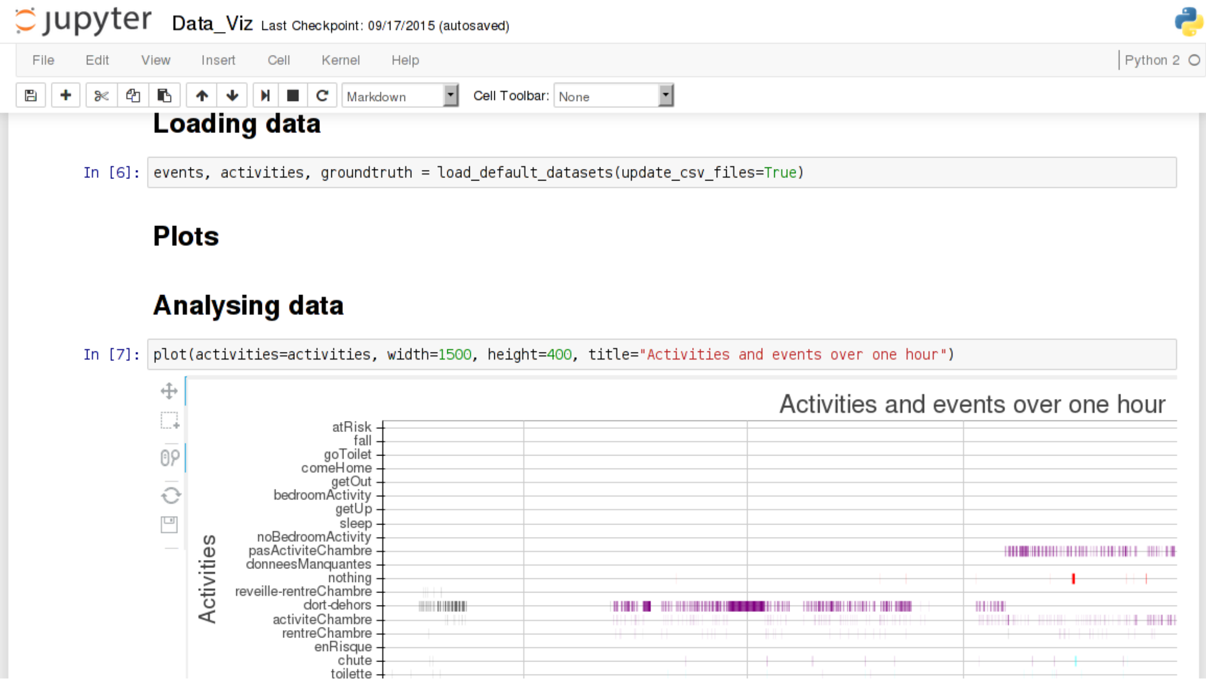


Figure 5.1: Jupyter Notebooks

```
knot
├── datasets
│     ├── events
│     └── activities
├── notebooks
├── knot
│     ├── data_analysis
│     ├── reasoning_engines
│     │        └── semantic_engine
│     ├── model
│     │     └── n3_files
│     ├── presets
│     │     ├── house.py
│     │     ├── data.py
│     │     └── reasoner.py
│     ├── simulator
│     ├── sinks
│     └── sources
├── readme.md
└── setup.py
```

Figure 5.2: File architecture of Knot

Loading The Library

Listing 5.2 illustrates a typical Knot code that loads the default datasets and creates a house.

```
# Import datasets and house presets
from reasoning.presets.dataset import load_default_datasets
from reasoning.presets import create_house

# Load and pre_process events, activities and groundtruth
# from datasets
events, activities, groundtruth = load_default_datasets(
    update_csv_files=True)
# Create a house with default setup
house = create_house('My House')
```

Listing 5.2: Loading Knot

Modular Architecture

Knot is designed to be modular. It is possible to create new modules for the following objects:

- Reasoning workflows;
- Conditions for the reasoning rules;
- Data sources;
- Output sinks;
- Sensors;
- Rooms;
- Items in the house.

The dynamic discovery of new modules is not available yet. At the moment, the developer of a new module needs to manually declare the module into the core of Knot.

5.3 Data Analysis

The primary motivation for the development of Knot was to process and visualize the data acquired from our deployment.

5.3.1 Getting Data From Database

Our datasets are extracted from UbiSMART, running in our deployment. We are interested in the *events* and an *activities* tables. They are saved in a Relational Database Management System (RDBMS) on a server, namely PostgreSQL. The first step for data analysis consists into exporting backups of the database into Comma-Separated Value (CSV) files, for more simplicity of use. This can be done through built-in functions in the RDBMS (Listing 5.3).

```
COPY event TO 'backup_event_<yyyy_mm_dd>.csv'\
  DELIMITER ',' CSV HEADER;
COPY activity TO 'backup_activity_<yyyy_mm_dd>.csv'\
  DELIMITER ',' CSV HEADER;
```

Listing 5.3: Exporting from Database to CSV format

5.3.2 Loading The Data Structure

Introducing The Pandas Library

Pandas is a library providing pandas data structures and data analysis tools for the Python programming language. It is built on top of Numpy, that is the fundamental package for scientific computing in Python.

Pandas provides two powerful models, namely *Series* and *Dataframes*. Series are vectors that can be indexed, with a particular focus on datetime index. Dataframes are matrix-like data structures, with features similar to the one of a table in relational databases. Powerful operations such as slicing, indexing, vectorial programming, etc. can be applied to Series and Dataframes.

Loading And Preprocessing Data

Once the dataset obtained in CSV format, I process the data using Python's Pandas. The first step is to cleanup the datasets by merging new data with existing ones, removing duplicates, reordering and renaming columns. At the moment, Knot includes the following functions for preprocessing:

- Rename sensors and activities;
- Rename residents, split residents by date (e.g., one room was occupied by resident A until 2015-01-01, then resident B);
- Remove data for a specific sensor of resident;
- Remove activities which have a duration of less than 1 second;
- Remove duplicates.

When loading datasets, Knot automatically looks for new CSV dump files from database, and integrates them into the main dataset. While the process can be customized at will, a preset has been defined to load default dataset from our deployment, and can be called in one line, through the `load_default_datasets` function.

The Dataframe Structures

Table 5.2 shows the structure of events and activities Dataframe in Knot.

		sensor	value
patient	date		
A	2014-09-23 13:37:00.000000	Bedroom Motion	On
	2014-09-23 14:05:36.000000	Main Door Sensor	Normal
	2014-09-23 14:05:53.000000	Main Door Sensor	Alert
	2014-09-23 14:05:55.000000	Main Door Sensor	Normal
	2014-09-23 14:05:57.000000	Main Door Sensor	Alert

(a) A Dataframe of events

		activity	position	since
patient	date			
A	2014-09-23 14:05:53.727	Busy	Bedroom	2014-09-23 09:01:23.041
	2014-09-23 14:06:53.825	Outside	Outside	2014-09-23 14:05:55.000
	2014-09-23 14:09:04.040	Come home	Bedroom	2014-09-23 14:07:06.000
	2014-09-23 14:15:10.000	Go Toilet	Toilet	2014-09-23 14:09:08.000
	2014-09-23 15:24:41.050	Outside	Outside	2014-09-23 15:12:10.000

(b) A Dataframe of activities

Table 5.2: Dataframes

5.3.3 Plotting Data

Introducing The Bokeh Library

Bokeh is a Python visualization library. It offers a simple syntax for plotting elegant and concise visualisations. In the Python scientific computing community, the most widely used library for data visualization is the legacy *Matplotlib* [7]. Yet, I found Bokeh to be more effective and to provide better outputs than Matplotlib. Bokeh is not not be confused with the Javascript's D3.Js library [8]; whereas we use D3.js for high quality visualisations in the UbiSMART framework, Bokeh offers more flexibility for rapid prototyping.

[7] `http://matplotlib.org/`
[8] `http://d3js.org/`

The Plot Method

Given events and activities Dataframes, Knot provides an advanced all-in-one `plot` function for visualizing data. The `plot` function's definition is given in Listing 5.4. Figure 5.3 shows example calls of the plot methods.

```
def plot(activities=None, events=None, groundtruth=None,
         patients=None, date_min=None, date_max=None,
         nb_columns=1, width=900, height=400, title=None,
         with_legend=True):
```

Listing 5.4: The Plot function

5.4 Reasoning Workflow

Knot includes its own reasoning engine, with the ability to run it over multiple sources of events (e.g., events dataframe). The reasoning engine is completely modular, and should not be limited to rule-based reasoning.

The original motivation to implement this reasoning engine was to run again the reasoner over existing datasets, and to test the improvements made over the original UbiSMART reasoner (Chapter 7).

5.4.1 The Reasoning Process

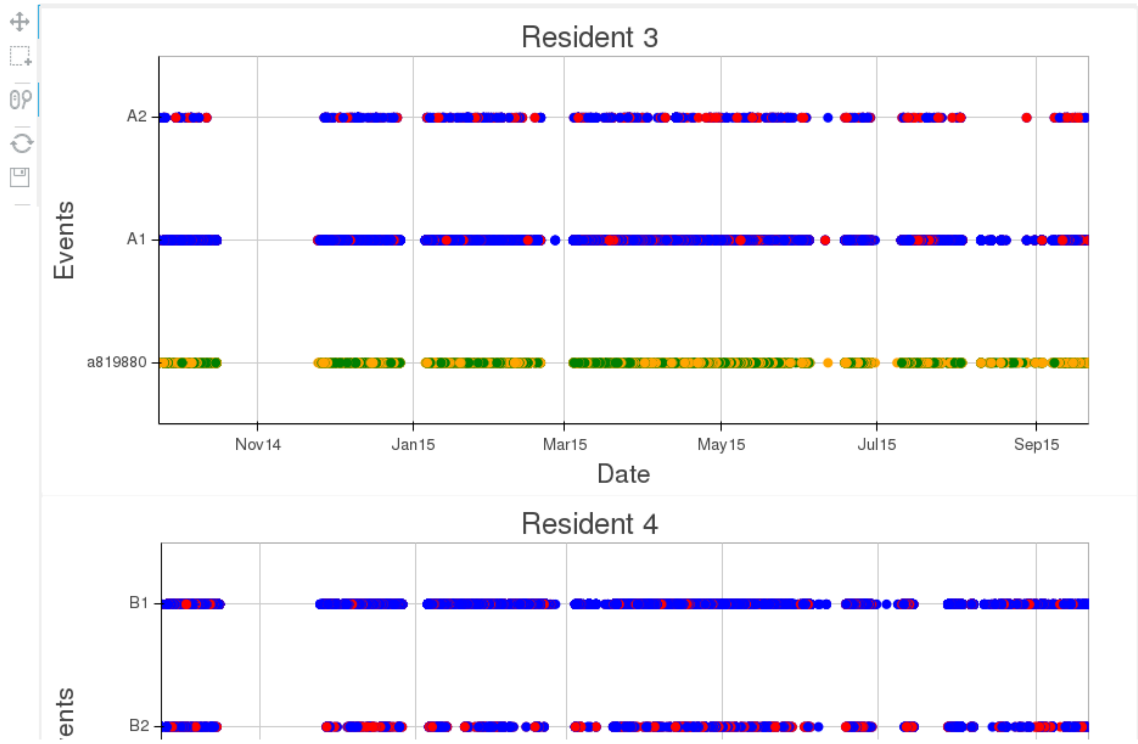
Introducing Coroutines

Knot's reasoning engine is built using a programming pattern named *coroutines*. It is important to understand coroutines before to tackle the reasoning process, as they typically include programming patterns that would be confusing otherwise (e.g., infinite loops).

A coroutine can be seen as an independent task, which accept an entry point, and can yield data continuously during their execution. It is a generalization of subroutines (i.e., function) that only accept one entry when their execution starts, and return data at the end of the execution.

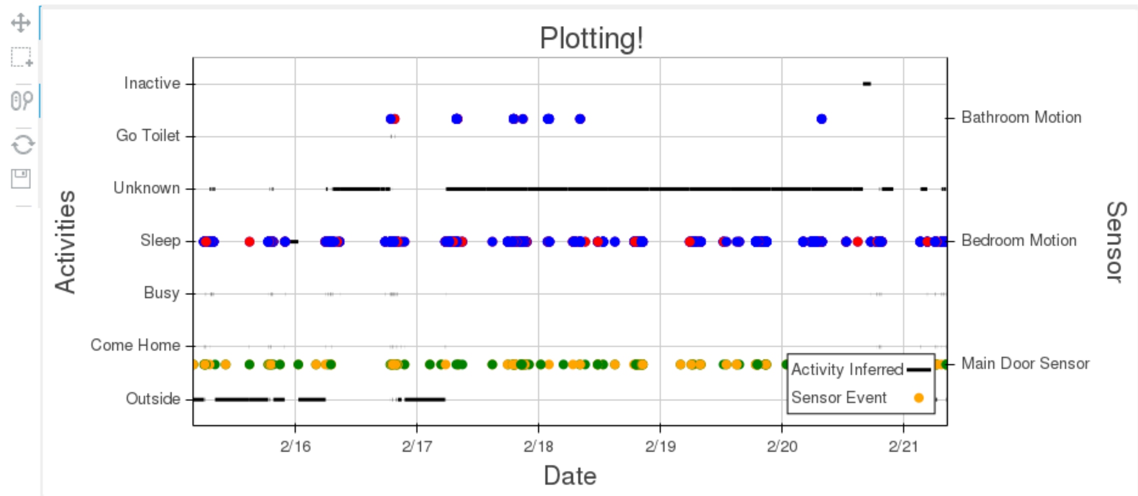
UNIX pipes have a behaviour equivalent to those of coroutines. Coroutines offer the advantage of developing semi-persistent (through the infinite loop) and mutually independent tasks, which can be assembled together at a higher-level logic. Listing 5.5 illustrates coroutines, and show how they can be implemented in python.

```
plot(events=events, patients=range(3,8))
```



(a) Plotting events for residents 3 to 8

```
plot(activities=activities,events=events, patients=['D'], date_min='2015-02-15', date_max='2015-02-22', title="Plotting!")
```



(b) Plotting events and activities for resident “D”

Figure 5.3: Illustrating the plot method

```
def sum_coroutine(target):
    target.next()
    while True:
        # wait here until a and b are received
        a, b = yield
        # send a + b to `receiver` (triggers the yield)
        target.send(a + b)

def receiver():
    while True:
        # wait here until a value is received
        value = yield
        print(value)

my_coroutine = sum_coroutine(receiver())
# in python, coroutines must in initiated with a first `next` call
my_coroutine.next()
for _ in range(2):
    a, b = (randint(0, 5), randint(0, 5))
    print("> %s+%s" % (a, b))
    # send a and b to `sum_coroutine` (triggers the yield)
    my_coroutine.send((a, b))
# >>> 2+0
# 2
# >>> 0+4
# 4
```

Listing 5.5: An example coroutine

It is interesting to notice that coroutines are also at the core of event-driven programming (i.e., NodeJS), and that Python has recently enhanced its support to coroutine in its versions 3.4 and 3.5.

High-Level Logic

Knot's reasoning process can be summarized as follows: a source produces events, which are sent to the reasoner, and the reasoner yields activities, which are sent to a sink. The process is built with coroutines, and could be summarized by the following pipe: `source | reasoner | sink`.

At the moment, these three tasks are all that is needed, but there would be no technical issue for inserting intermediate tasks to the process.

The Reasoning Engine

The reasoner itself is composed of several steps. Steps are subroutines (i.e., normal functions), not coroutines. Every reasoning steps are executed sequentially during each iteration (Listing 5.6).

```
def reason(self, steps, target):
    while True:
        # when a new status is sent (i.e., a sensor event)
        status = yield
        # We run each step sequentially
        for step in steps:
            status = step(status)
        # We send the final status (i.e., an activity)
        target.send(status)
```

Listing 5.6: The reasoner coroutine

This process is reasoner-agnostic, and works as well for rule-based reasoning or for any data-driven method. In this document, I will focus on the implementation of rule-based reasoning. Also, whereas Knot may not use an ontological *reasoning*, it always relies on an ontological *model* in the back-end.

Sources

Sources are events producers. I have explored four different sources:

Dataframe source Yield events one by one from an event Dataframe (Listing 5.7);

Simulator source Yield events as they are generated by the simulator (Section 5.5);

Socket source Yield events sent through a network socket;

Standard input source Yields event as the developers writes them down manually (for debugging purpose).

```
def dataframe_source(target, events):
    for event in events.values:
        # event: [date, sensor, value]
        target.send(event)
```

Listing 5.7: The Dataframe source

Sinks

Sinks are coroutines which receive, but do not produce any content. Knot's sinks subscribe to activities. The default sink is the Dataframe Sink, which saves inferred activities in a Dataframe. An issue is that a coroutine does not end properly, so it can not return the resulting Dataframe as a final value; and the Dataframe cannot be accessed outside of the coroutine's scope. The Solution was to embed the coroutine inside a class, and to make the Dataframe as an attribute of the class instance (Listing 5.8).

```
class DataframeSink(object):
    def __init__(self):
        self.activities = pd.DataFrame(columns=[
            'house', 'date', 'activity', 'position', 'since'
        ])

    def __call__(self, house):
        self.co = self.sink(house)
        return self

    def send(self, *args):
        self.co.send(*args)

    @coroutine
    def sink(self, house):
        self.activities = self.activities.append(
            activity, ignore_index=True)
```

Listing 5.8: The Dataframe sink

5.4.2 Implementing The UbiSMART Reasoning

The UbiSMART reasoner relies on the dynamic environment introduced in Section 8.2. The Euler Rule Engine is executed on the generated rules, towards the generated house. It is composed of the following steps, documented in the thesis of Tiberghien [31]:

- Save event into the ontology;
- Estimate motion in rooms;
- Reason on low-level context;
- Reason on activities;
- Read Activities, and yield it to the ontology.

5.5 Simulator

An AAL simulator was created in response to a need to quickly generate a large number of events, and to have full control on it. It addresses the three following use cases:

– Generate realistic sensor events on-demand;

– Validate that the reasoning engine correctly infers an activity, when a "canonical" scenario of this activity is executed;

– Simulate sensors before we integrate them in the physical world.

To use the simulator, the developer must first describe the environment, using the houses factory (Section 8.2.1). The developer then defines a scenario, that consists into a sequence of actions to be executed by the resident. Once the simulator is executed, the generated sensor events are saved into a Dataframe, or used as a source for the reasoning engine (Section 5.4.1).

Several similar simulator exist in the literature [313, 314], with more advanced feature that the one described in this document. But the Knot's simulator distinguishes itself because it is based on an ontological model, which is shared with the reasoning engine, allowing for a symmetry between simulation and reasoning.

Introducing Simpy And Discrete Event Simulation

Simpy is a process-based discrete-event simulation framework written in Python. Simpy can either perform simulations "as fast as possible", simulating real time, or by manually stepping through the events.

According to [315], a discrete-event simulation models the operation of a system as a discrete sequence of events in time. Each event occurs at a particular instant in time and marks a change of state in the system. Discrete event simulation is thus a fast and elegant method to simulate the behaviour of a system over a period of time.

5.5.1 The Objects

The simulator integrates the objects included in Knot, namely:

– Person;

– Room;

– Door;

– Object;

– Sensor.

Each object has a certain sets of actions that can be triggered. Objects are interconnected, so that when an action is triggered, it may interact with other objects, and trigger new actions. To initiate the process, a scenario includes a sequence of actions to be performed by the simulated resident.

The Resident

A resident — and more generally a person — has three possible actions at the moment: *go to room*, *keep moving in the room* and *stay immobile*. Interactions between a resident and specific objects (e.g., bed) has not yet been implemented. The *keep moving in the room* action accepts a duration, and a frequency of motion (e.g., one noticeable motion every second). *Stay immobile* is a specialization of *keep moving in the room*, where the user moves at a frequency of 0.

To go to a specific room in the simulator, the resident does not teleport himself. He moves realistically among rooms (given that doors have been defined) towards his destination, staying a certain number of seconds in each room (by default, 8 seconds). To do so, the simulator uses a Breadth-First Search [125] in a non-ponderated graph of rooms, with the doors playing the role of edges.

When the resident moves in a room, he triggers the *motion* action to the room object. When he moves across a door, he will trigger an open/close action to the door object (although this may be unrealistic where doors remain open, and should be further calibrated).

Rooms, Doors And Items

At the moment, there was no item implemented besides rooms and doors. The behaviour of rooms and doors is extremely simple: when a motion happens in the room, or when the door is open or closed, these objects will notify the sensors attached to it.

The Sensors

The simulator includes two sensors: *door sensor* and *Motion Sensor*.

When a door is open (resp. closed), the `notify_open` method (resp. `notified_closed`) is called, and a sensor event “ALERT” (resp. “NORMAL”) is yielded, consistently with the physical behaviour of these sensors.

Motion sensors are more complex. They send “ON” when a motion is observed. While there is motion, “ON” signals are sent every 10 seconds. After 60 seconds without motion, a “OFF” signal is sent. Listing 5.9 shows the implementation of this behaviour.

```
ON_DELAY = 10
OFF_DELAY = 60

latest_event = 0
sensor_resource = PreemptiveResource(env, capacity=1)
on_process = None
# Runs the off_process on init
env.process(_process_off())

def notify_motion(new_motion):
    """Triggered by a new motion in the associated room."""
    # If there is no currently running _process_on(), we run it
    if on_process is None or on_process.triggered:
        on_process = env.process(_process_on())
        latest_event = env.now

def _process_on():
    # ON process steals the priority to the ongoing OFF process
    with sensor_resource.request(priority=1) as req:
        yield req
        yield env.timeout(ON_DELAY)

def _process_off():
    # Runs continually, loops when interrupted
    while True:
        with sensor_resource.request(priority=2) as req:
            try:
                yield req
                #   Wait OFF_DELAY seconds
                off_time = latest_event + OFF_DELAY
                yield env.timeout(off_time - env.now)

                send_off_signal()
                #   Wait until something happens
                yield env.timeout(float('inf'))
            except simpy.Interrupt:
                # When a ON process interrupts the OFF process
                continue
```

Listing 5.9: Simulating a motion sensor

5.5.2 Example

Following, I illustrate the use of the simulator through a complete example.

I will refer to the house in Figure 5.4, defined using the houses factory (Section 8.2.1)

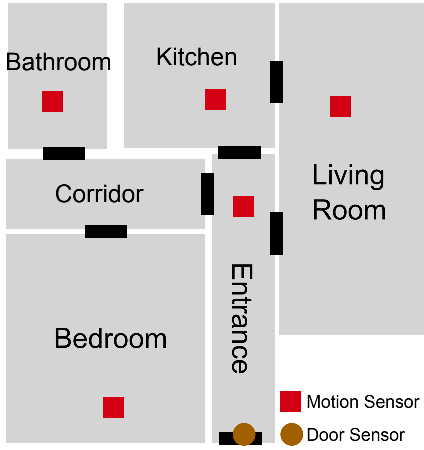


Figure 5.4: House to be simulated

I define a scenario where the resident will go to the kitchen, move for a while, spend a few minutes in the living room, then go to bathroom, and finally go to the bed and stay immobile for two hours (Listing 5.10).

```
@scenario
def scenario(simulator, env, actor, house):
    actor.set_initial_position("Living Room")
    simulator.set_clock(time)
    yield env.process(actor.go_to("Kitchen"))
    yield env.process(actor.keep_moving(timedelta(minutes=1)))
    yield env.process(actor.go_to("Living Room"))
    yield env.process(actor.keep_moving(timedelta(minutes=3)))
    yield env.process(actor.go_to("Bathroom"))
    yield env.process(actor.keep_moving(timedelta(minutes=2)))
    yield env.process(actor.go_to("Bedroom"))
    yield env.process(actor.keep_moving(timedelta(seconds=30)))
    yield env.process(actor.stay_immobile(timedelta(hours=2)))
```

Listing 5.10: Example scenario for the simulator

After executing the scenario (Listing 5.11), we obtain the events in Table 5.3.

```
time = parse("2015-09-20 10:00:00")
simulator = Simulator(house, init_time=time)
simulator.run(scenario)
```

Listing 5.11: Executing the scenario with the simulator

Date	Sensor	Value
2015-09-20 10:00:02	Living Room Motion	On
2015-09-20 10:00:12	Living Room Motion	On
2015-09-20 10:00:22	Living Room Motion	On
2015-09-20 10:00:32	Living Room Motion	On
2015-09-20 10:00:42	Living Room Motion	On
2015-09-20 10:00:52	Living Room Motion	On
2015-09-20 10:01:02	Living Room Motion	On
2015-09-20 10:01:12	Living Room Motion	On
2015-09-20 10:01:18	Entrance Motion	On
2015-09-20 10:01:34	Bathroom Motion	On
2015-09-20 10:01:46	Bathroom Motion	On
2015-09-20 10:01:58	Bathroom Motion	On
2015-09-20 10:02:10	Bathroom Motion	On
2015-09-20 10:02:12	Living Room Motion	Off
2015-09-20 10:02:18	Entrance Motion	Off
2015-09-20 10:02:22	Bathroom Motion	On
2015-09-20 10:02:38	Bedroom Motion	On
2015-09-20 10:02:48	Bedroom Motion	On
2015-09-20 10:02:58	Bedroom Motion	On
2015-09-20 10:03:08	Bedroom Motion	On
2015-09-20 10:03:22	Bathroom Motion	Off
2015-09-20 10:04:08	Bedroom Motion	Off

Table 5.3: Simulated events

5.5.3 Opening

The simulator presented in this section is considered a useful addition to this thesis work, and not a keystone. It has been developed in response to a specific need, and there was no reason to make it more advanced at the moment. Yet, powerful extensions have been considered:

– In the future, it would be interesting to integrate stochastic simulation, in order to obtain various datasets by running the same scenarios several times, and to observes under which conditions the reasoner will fail.

– So far, the interactions between the residents and objects such as bed are not implemented. It would become dramatically more powerful after implement these interactions. I envision a repository of objects and actions to be plugged to the simulator, to create richer and more realistic simulations.

– Another opportunity would be to integrate higher-level behavioral simulation, that would simulate the resident's activities over a day, without the need for predefined scenarios. A similar research has been successfully taken by Helal et al. [313].

5.6 Discussions

Through this section, I have introduced the Knot prototyping framework for Activity Recognition. This framework includes a large number a features related to Activity Recognition: analyzing datasets, create environment and rules dynamically, edit the reasoning engine and run it on various sources, and use a simulator to generate realistic events. These features have been useful for the research work described in Chapters 7 and 8.

Knot is validated in Chapter 9, through a complete use-case.

5.6.1 Limitations and Perspectives

Prototyping Activity Recognition is a large topic, which I consider to be in its early days. Working on improving this process does feel like opening the pandora's box. Numerous limitations have been founds. Among these limitations, the followings have kept my attention.

– Although Knot can run the reasoning on past events, the reasoning process is extremely slow, due to the Euler Rule Engine.

– Knot is not integrated with UbiSMART yet. Current deadlocks are that Knot does not handle multiple houses running simultaneously, and is not yet ready for real time.

– Using RDFLib is an enhancement from modifying N3 files manually. But the relation between the Python model and the RDF back-end is rather confusing. The use of an Object-Relational Mapping (ORM) (e.g., RDFalchemy) would allow for a more elegant implementation, and has been further developed in Chapter 8.

– Knot is modular; but for a wide adoption we would expect a real system of plug-ins, where AAL application developers would never have to work on the core.

Part IV

A Methodology for Activity Recognition

Life doesn't make any sense without interdependence. We need each other, and the sooner we learn that, the better for us all.

Erik Erikson

6 Location Tracking and Multiuser Detection in Real Condition

6.1 Problematic

Figure 6.1 presents a situation where certainly more than one person is active in the residence. These situations are called "multiuser situations". The resident is first assumed to be alone, active in the kitchen (zone A). The main door is open, sensors observe movement on the corridor, and the door is closed again (Zone B). Then, the two persons are probably in the same room, as only one room is active (zone C). We then observe conflicting sensor events, where two rooms are active at the same time (zone D). Finally, the door opens and closes again (zone E), and we observe simple sensor events over one room at a time. We may assume that the resident is now alone (Zone F).

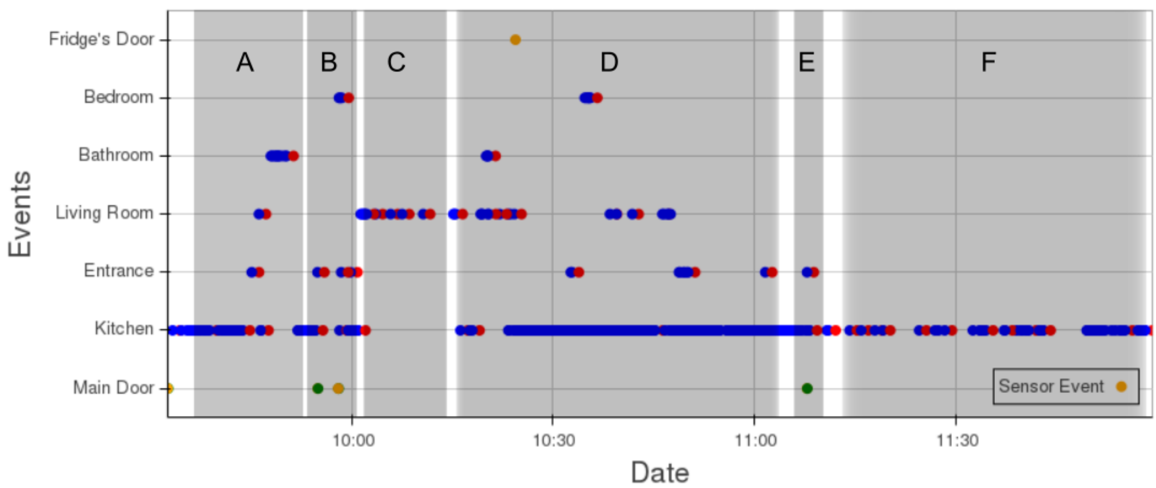


Figure 6.1: A Multiuser Situation

The former observations have been done manually. In terms of data, multiuser situations generate interwoven activities, and overlapping sensor events between rooms, which are extremely hard to process. Fortunately, there is no urge for the system to recognize activities when several persons are present in the residence. Indeed, the main purpose of activity recognition is to detect risky activities; one may safely consider that if a resident has visitors, the visitors will be able to react to a danger. The system can then deactivate itself for the time of the multiuser situation.

Therefore, the system is required to detect whether or not we are in a multiuser situation.

In this chapter, I introduce an algorithm to detect where is the resident located, or when there is a multiuser situation. Using this algorithm, the situation observed in Figure 6.1 will result in the insights seen in Figure 6.2.

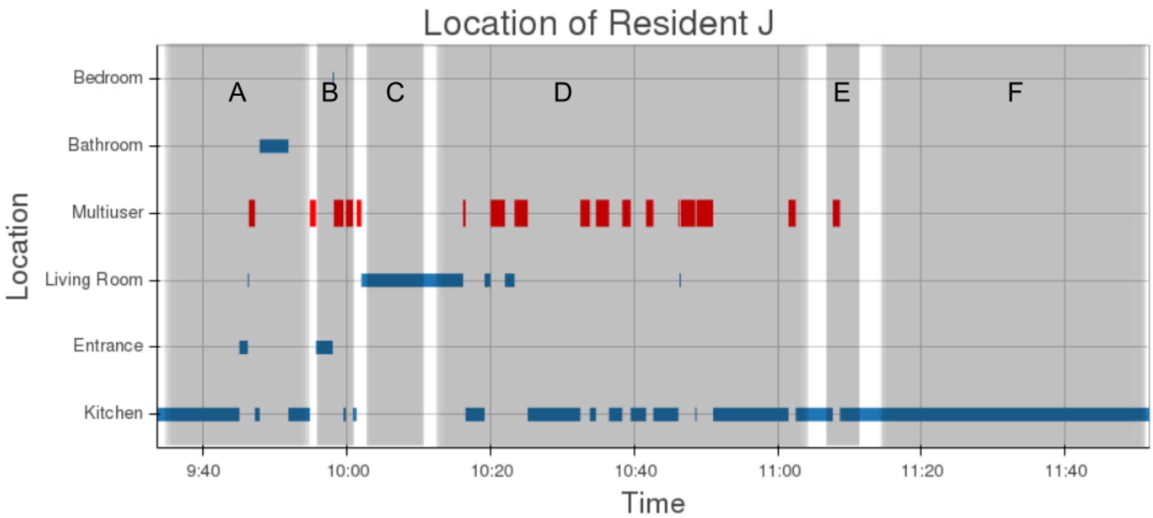


Figure 6.2: Automatic detection of multiuser situations

6.2 Related Works

Several researchers have identified the important role that can be played by social interactions in a AAL system. In consequence, multiuser detection, social interaction analysis, and Activity Recognition in multiuser situations is an active focus point in AAL. This research yields rich outcomes when combined with wearable sensors [236, 316], smartphones [317] or depth vision using Kinect [318]. Ye et al. [319] propose a high-level semantic model to perform Activity Recognition with concurrent activities, which can apply to multiuser situations.

As for environmental sensors, Cook et al. [320] infer indoor location and analyze social interactions using Bayesian methods on a rich set of environmental sensors; they obtain an accuracy of 85%. Aicha et al. [321] have performed an in-depth study on Activity Recognition in multiuser situations using simple environmental sensors and HMM. They also obtain an accuracy of 85%.

I believe that in our case, Activity Recognition in multiuser situations is out of our requirement, and I have chosen to focus solely on detecting these multiuser situations. By narrowing the scope, the method I propose in this chapter can be executed with as little as a motion sensor in each room.

6.3 Detecting Multiuser Situations

I have created an algorithm that performs location tracking and multiuser detection [295], based on concurrent sensor activations. I have chosen a data-driven approach, in order to efficiently process the data. A knowledge-driven approach has been considered, but this algorithm does not require the richness of model brought by ontological reasoning. For brevity purpose, the implementation details have been separated from this chapter, and can be found in Appendix B.

6.3.1 Initial Dataset

As input, the algorithm receive a sequence of sensor events, where each sensor event was triggered by a movement in a room, or by the absence of movement in that room for 60 seconds. At the moment, we focus solely on motion sensors, and ignore any other sensors (e.g., door sensors). Events are defined as a tuple $< house; date; room; value >$ (Table 6.1). In the current implementation, the sequence of events is implemented is a dataframe.

6.3.2 Reformatting the Dataset

As a first step, the algorithm groups the sequence of events by rooms (Listing B.1 from Appendix B). We obtain Table 6.2. The interest of grouping events by rooms, is that it becomes possible to apply a function independently to each room.

The second step consists in making events as a $i \times j$ matrix, where i is the room, and j the date, in order to obtain Table 6.3). This step is displayed in Listing B.2 from Appendix B.

This step requires to resample events with a uniform frequency for every rooms, ranging from the first event's time to the last event's time. This way, every rooms share the same temporal index. I define the resampling period to 10 seconds, since sensors are specified to send events at most every 10 seconds. To fill-in the blanks, we use the Last Observation Carried Forward (LOCF) algorithm. The dataframe is then reshaped into a matrix where each room becomes a column, as shown in Table 6.3.

House	Date	Sensor	Value
A	2014-11-24 17:13:25	Living Room	On
	2014-11-24 17:13:27	Living Room	On
	2014-11-24 17:13:29	Kitchen	On
	2014-11-24 17:13:31	Living Room	On
	2014-11-24 17:13:37	Kitchen	On
	2014-11-24 17:13:44	Kitchen	On
	2014-11-24 17:13:51	Kitchen	On
	2014-11-24 17:14:06	Bedroom	Off
	2014-11-24 17:14:27	Living Room	Off
	2014-11-24 17:14:29	Living Room	Off
	2014-11-24 17:14:55	Kitchen	Off
	2014-11-24 17:15:07	Kitchen	On
	2014-11-24 17:15:36	Kitchen	On
	2014-11-24 17:15:53	Kitchen	On
	2014-11-24 17:16:05	Kitchen	On
	2014-11-24 17:16:30	Bedroom	On
	2014-11-24 17:16:42	Bedroom	On
	2014-11-24 17:16:49	Bedroom	On
	2014-11-24 17:16:56	Bedroom	On
	2014-11-24 17:17:04	Kitchen	On

Table 6.1: Original Dataset of events

House	Sensor	Date	Value
A	Bedroom	2014-11-24 17:14:06	Off
		2014-11-24 17:16:30	On
		2014-11-24 17:16:42	On
		2014-11-24 17:16:49	On
		2014-11-24 17:16:56	On
	Kitchen	2014-11-24 17:13:29	On
		2014-11-24 17:13:37	On
		2014-11-24 17:13:44	On
		2014-11-24 17:13:51	On
		2014-11-24 17:14:55	Off
		2014-11-24 17:15:07	On
		2014-11-24 17:15:36	On
		2014-11-24 17:15:53	On
		2014-11-24 17:16:05	On
		2014-11-24 17:17:04	On
	Living Room	2014-11-24 17:13:25	On
		2014-11-24 17:13:27	On
		2014-11-24 17:13:31	On
		2014-11-24 17:14:27	Off
		2014-11-24 17:14:29	Off

Table 6.2: Events grouped by sensors

House	Date	Bedroom	Kitchen	Living Room
	2014-11-24 17:13:20	NaN	On	On
	2014-11-24 17:13:30	NaN	On	On
	2014-11-24 17:13:40	NaN	On	On
	2014-11-24 17:13:50	NaN	On	On
	2014-11-24 17:14:00	Off	On	On
	2014-11-24 17:14:10	Off	On	On
	2014-11-24 17:14:20	Off	On	Off
	2014-11-24 17:14:30	Off	On	Off
	2014-11-24 17:14:40	Off	On	Off
	2014-11-24 17:14:50	Off	Off	Off
	2014-11-24 17:15:00	Off	On	Off
A	2014-11-24 17:15:10	Off	On	Off
	2014-11-24 17:15:20	Off	On	Off
	2014-11-24 17:15:30	Off	On	Off
	2014-11-24 17:15:40	Off	On	Off
	2014-11-24 17:15:50	Off	On	Off
	2014-11-24 17:16:00	Off	On	Off
	2014-11-24 17:16:10	Off	On	Off
	2014-11-24 17:16:20	Off	On	Off
	2014-11-24 17:16:30	On	On	Off
	2014-11-24 17:16:40	On	On	Off
	2014-11-24 17:16:50	On	On	Off
	2014-11-24 17:17:00	On	On	Off

Table 6.3: Events resampled to uniform frequency, per room

6.3.3 Inferring Presence and Movement of the Resident in Rooms

Once the pre-treatment is completed and the dataset is properly formatted, we want to detect the presence of the resident in rooms. Indeed, motion sensors only send binary data, whether there is motion in a room or not. There are two use cases for the absence of motion: the resident may be absent, or immobile. Thus, rather than binary On/Off events, we would like to obtain the status of a room among the following values: *Absent*, *Immobile*, or *Movement*.

To infer this information, we apply a set of rules to each element of the matrix (Listing B.3) from Appendix B:

- When a sensor sends a "On" signal in room X, it indicates a *Movement* in room X.
- when a sensor sends a "Off" signal in room X, if the resident is present in room Y (*Immobile* or in *Movement*), he is Absent from room X.
- When a sensor sends "Off" signal in room X, if the resident is not currently seen in any other room, there are two cases:
  - If the resident was last seen in room X, we consider that he is still there, and infer *Immobile* in room X.
  - If the resident was last seen in another room, we consider that he is *Absent* from room X.

Through these rules, we obtain a more meaningful dataset, showing the actual presence of the resident (Table 6.4).

6.3.4 Detecting Multiuser Situations

The next step is critical. We want to detect multiuser situations, when the resident receives visit. Given the matrix in Table 6.4, detecting multiuser situations is straightforward. The system checks whether more than one room indicates presence at a specific time (i.e., *Immobile* or *Movement*); when it is the case, it infers a multiuser situation (Listing B.4). From the past sequence, we observe two occurrences of multiuser (Table 6.5).

6.3.5 Final Result: Location Tracking

At this stage of the process, we have essentially all the informations we require. Yet, we would like the algorithm to refactor these informations, and provide the location of the resident at any given time.

The next step consists into concluding on the resident's location. To do so, we handle two cases for every rows of the matrix (Listing B.5 from Appendix B):

- If there is no multiuser situation, we set the resident's location as the room being activated (i.e., *Movement* or *Immobile*).

House	**Date**	**Bedroom**	**Kitchen**	**Living Room**
	2014-11-24 17:13:20	Unknown	Movement	Movement
	2014-11-24 17:13:30	Unknown	Movement	Movement
	2014-11-24 17:13:40	Unknown	Movement	Movement
	2014-11-24 17:13:50	Unknown	Movement	Movement
	2014-11-24 17:14:00	Absent	Movement	Movement
	2014-11-24 17:14:10	Absent	Movement	Movement
	2014-11-24 17:14:20	Absent	Movement	Absent
	2014-11-24 17:14:30	Absent	Movement	Absent
	2014-11-24 17:14:40	Absent	Movement	Absent
	2014-11-24 17:14:50	Absent	Immobile	Absent
	2014-11-24 17:15:00	Absent	Movement	Absent
A	2014-11-24 17:15:10	Absent	Movement	Absent
	2014-11-24 17:15:20	Absent	Movement	Absent
	2014-11-24 17:15:30	Absent	Movement	Absent
	2014-11-24 17:15:40	Absent	Movement	Absent
	2014-11-24 17:15:50	Absent	Movement	Absent
	2014-11-24 17:16:00	Absent	Movement	Absent
	2014-11-24 17:16:10	Absent	Movement	Absent
	2014-11-24 17:16:20	Absent	Movement	Absent
	2014-11-24 17:16:30	Movement	Movement	Absent
	2014-11-24 17:16:40	Movement	Movement	Absent
	2014-11-24 17:16:50	Movement	Movement	Absent
	2014-11-24 17:17:00	Movement	Movement	Absent

Table 6.4: Presence of the resident in rooms

House	Date	Bedroom	Kitchen	Living Room	Multiuser
A	2014-11-24 17:13:20	Unknown	Movement	Movement	✓
	2014-11-24 17:13:30	Unknown	Movement	Movement	✓
	2014-11-24 17:13:40	Unknown	Movement	Movement	✓
	2014-11-24 17:13:50	Unknown	Movement	Movement	✓
	2014-11-24 17:14:00	Absent	Movement	Movement	✓
	2014-11-24 17:14:10	Absent	Movement	Movement	✓
	2014-11-24 17:14:20	Absent	Movement	Absent	✗
	2014-11-24 17:14:30	Absent	Movement	Absent	✗
	2014-11-24 17:14:40	Absent	Movement	Absent	✗
	2014-11-24 17:14:50	Absent	Immobile	Absent	✗
	2014-11-24 17:15:00	Absent	Movement	Absent	✗
	2014-11-24 17:15:10	Absent	Movement	Absent	✗
	2014-11-24 17:15:20	Absent	Movement	Absent	✗
	2014-11-24 17:15:30	Absent	Movement	Absent	✗
	2014-11-24 17:15:40	Absent	Movement	Absent	✗
	2014-11-24 17:15:50	Absent	Movement	Absent	✗
	2014-11-24 17:16:00	Absent	Movement	Absent	✗
	2014-11-24 17:16:10	Absent	Movement	Absent	✗
	2014-11-24 17:16:20	Absent	Movement	Absent	✗
	2014-11-24 17:16:30	Movement	Movement	Absent	✓
	2014-11-24 17:16:40	Movement	Movement	Absent	✓
	2014-11-24 17:16:50	Movement	Movement	Absent	✓
	2014-11-24 17:17:00	Movement	Movement	Absent	✓

Table 6.5: Multiuser situations

– Else, if there is a multiuser situation, we set the location as *Multiuser*.

This process results in the list of locations shown in Table 6.6.

House	Date	Location
A	2014-11-24 17:13:20	Multiuser
	2014-11-24 17:13:30	Multiuser
	2014-11-24 17:13:40	Multiuser
	2014-11-24 17:13:50	Multiuser
	2014-11-24 17:14:00	Multiuser
	2014-11-24 17:14:10	Multiuser
	2014-11-24 17:14:20	Kitchen
	2014-11-24 17:14:30	Kitchen
	2014-11-24 17:14:40	Kitchen
	2014-11-24 17:14:50	Kitchen
	2014-11-24 17:15:00	Kitchen
	2014-11-24 17:15:10	Kitchen
	2014-11-24 17:15:20	Kitchen
	2014-11-24 17:15:30	Kitchen
	2014-11-24 17:15:40	Kitchen
	2014-11-24 17:15:50	Kitchen
	2014-11-24 17:16:00	Kitchen
	2014-11-24 17:16:10	Kitchen
	2014-11-24 17:16:20	Kitchen
	2014-11-24 17:16:30	Multiuser
	2014-11-24 17:16:40	Multiuser
	2014-11-24 17:16:50	Multiuser
	2014-11-24 17:17:00	Multiuser

Table 6.6: Location of the resident

Finally, we want to compress the information and merge consecutive rows in the same location. To do so, rather than saving punctual informations (i.e., the location at a given time), we save the resident's location — or multiuser situation — into a time window (i.e., from …until …). This is implemented in Listing B.6 from Appendix B. The final result is present in Table 6.7.

House	Date	Until	Location
A	2014-11-24 17:13:20	2014-11-24 17:14:20	Multiuser
	2014-11-24 17:14:20	2014-11-24 17:16:30	Kitchen
	2014-11-24 17:16:30	NaT	Multiuser

Table 6.7: Final dataset — location of the resident

6.4 Discussions

In this Chapter, I have introduced an algorithm to track the location of the resident, and detect multiuser situations. This method does not claim to be more accurate than other methods developed in the literature, but it is certainly more lightweight and does not require any advanced sensors. This algorithm is validated in Chapter 9, where it is applied against our entire dataset in a house.

As a direct benefit of detecting multiuser situations, our system can stop performing Activity Recognition as soon as such a situation is detected. On a more long-term point-of-view, it may also be an interesting insight to detect loneliness in elderly people's life, or to measure a decrease in social activities over time though statistical analysis, which may indicate a deterioration of the person's health.

A limitation of this algorithm is that it is limited to motion sensors. If the resident and his visitors stay in the same room, the system will tend to detect them as a single user. As a perspective, this issue could be partially addressed by integrating the main door, given that a multiuser situation does not start nor end without the door being opened.

The doorstep to the temple of wisdom is a knowledge of our own ignorance.

Benjamin Franklin

7 A Method for Improving Activity Recognition Reasoning

7.1 Introduction

Context-awareness and Activity Recognition are necessary to trigger adequate services on appropriate media, given a certain situation/context. An example can be to send an alert when a risk/dangerous situation is detected. Thus, faulty inferences (i.e., lacks of Accuracy and Precision) in Activity Recognition are among the most critical issues I have identified in my experience with Ambient Assisted Living (AAL) for elderly people. An Inaccurate system will generate a number of misleading reactions, whereas an Imprecise system will possibly be Accurate, but hardly useful. Following I discuss my experience to illustrate the Accuracy and Precision issues [322].

7.1.1 My Experience with Ambient Assisted Living

In this section, I focus on three lessons learned from our real deployment, that can be summarized as follows:

1. **Feedback in the Development Process** *Simulations and laboratory trials are an incredible asset for prototyping of AAL solutions, because they can generate data on-demand. But they should always lead to real applications. Only through real applications one can witness the limitations, Accuracy and Precision issues of the system in unexpected situations.*

   Among the unexpected behaviors I have observed, I noticed that certain inferences in our reasoner were particularly Inaccurate, whereas others were Accurate but too Imprecise to be useful.

   The feedback on the real use of the system has to be integrated in the iterative development process, not only in the assessment phase, but also in the system improvement phase.

2. **Lower Deployment Cost** *AAL solutions should be offered at a reasonable price, with a minimal installation cost. They also should be adaptable to divers environments and be able to use on-shelf products, thus there is a need to simplify the deployment process.*

   We have chosen to simplify the deployment, by using simple and generic sensors as much as possible. Thus, under these conditions, the activity recognition becomes more complex and more error-prone. The need for a feedback loop, that enables to observe failures in the reasoning process and use them as a source of improvement, is even more critical.

3. **User in the Loop** *Activity recognition is a mean, not an end. It is only helpful if it contributes to user experiences through useful services (e.g., his safety).*

   Our experience in real deployments and the insights we gathered from a survey we conducted led us to believe that the most valuable criteria for users are *helpfulness*, *usability*, *security*, *acceptability*, *maintainability*, *reliability*, *quality of the user experience* and *accuracy* [1]. In accordance with this assertion, I target in this section the two criterias of *Accuracy* and *Precision*.

These three lessons were my motivation to improve the flexibility, the accuracy, and eventually the usefulness of Activity Recognition. Therefore, I propose a process that enables improving the Activity Recognition outcome. In this process, I first introduce a metric to measure the Accuracy of a reasoning engine. Then, I integrate this measure of Accuracy into the reasoning process. This integration of the Accuracy enables to converge, in the most reasonable way, towards a more reliable service delivery. I consider a system to be reasonable if it selects the best option available given its knowledge of the environment; which means that the system should be as Precise as possible, while still Accurate.

The Accuracy issue has also been highlighted in other researches on activity recognition. Several researchers have been interested in the Accuracy using the datasets gathered from the real-world deployment [237, 298, 323]. For example, Cook [298] has used CASAS datasets to perform machine-learning methods for activity recognition; the results of her team were around 75% on rich training datasets. Kadouche et al. used SVM for activity recognition and they obtained an Accuracy of 88% [323]. Chung et al. [237] applied activity recognition in an application targeting nursing home. Using Hierarchical Context Hidden Markov Model (HC-HMM), they obtain a recognition Accuracy of 85%. Nevertheless, their activity recognition process relies on cameras, which is often associated with acceptability issues.

To my knowledge, the Precision has not been addressed in other works.

I believe that Accuracy and Precision are key criteria in the validation process of any activity recognition, since the Accuracy is an indicator of the reliability of an AAL system and the Precision is an indicator of its usefulness. I also believe that using these two indicators in an iterative process of activity recognition will even enable quick convergence of the system.

For clarity purpose, it is important to distinguish the concepts of *Precision* and *Accuracy*. I define the Precision of activity recognition by a measurement of the granularity of the detected activities; for instance a high Precision would be "on phone with his daughter," and a low Precision would be "in the bedroom." On the other hand, I define Accuracy as the confidence of having a correct inference (i.e., the confidence that the inferred activity matches the reality). An Accuracy of 99% would be considered highly Accurate, whereas an Accuracy of 50% would be considered Inaccurate.

Structure Of This Chapter

In this chapter I present a method which introduces Accuracy and Precision in the activity recognition process. Section 7.2 illustrates the problems of fault inferences. Section 7.3 introduces a method to quantify the Accuracy of a reasoning engine. Section 7.4 discusses a method to optimize the decision-making. Section 7.5 describes how this method has been integrated with our current technology stack. Section 7.7 introduces thrilling perspectives.

7.2 Precision and Accuracy Issues in Activity Recognition

In this section, I illustrate the Accuracy and Precision issues through examples from our real-world deployment of UbiSMART, either in an individual residence or in the nursing home, in order to highlight the importance of the subject. The output of our system is sensor event data (dots in the figures) and inferred activities (vertical lines in the figures). Only visualizing the sensor events, we are able to detect certain emerging intuitive patterns, such as the night sleep in Figure 7.1 and taking lunch in Figure 7.2.

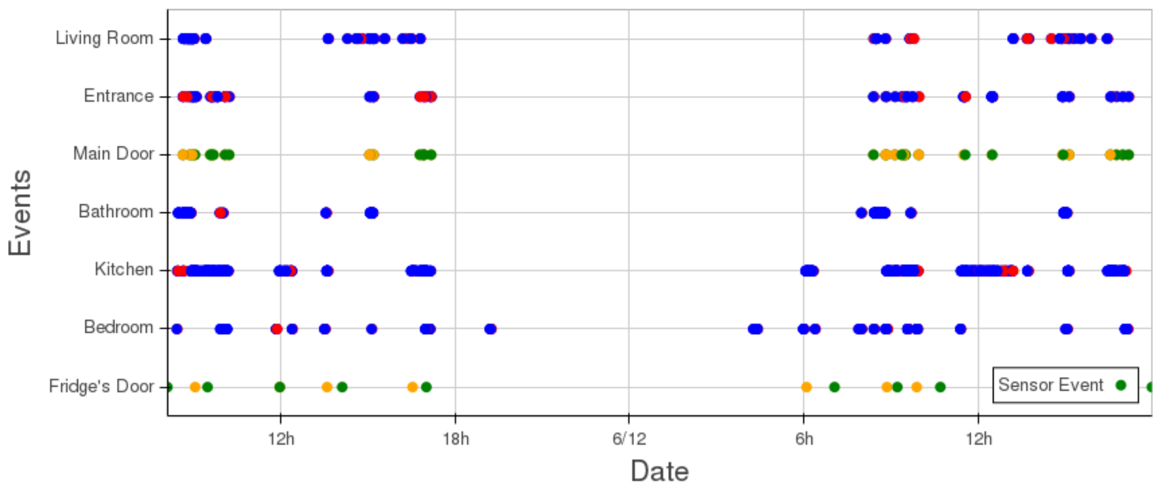


Figure 7.1: Sensor events over a day in a house

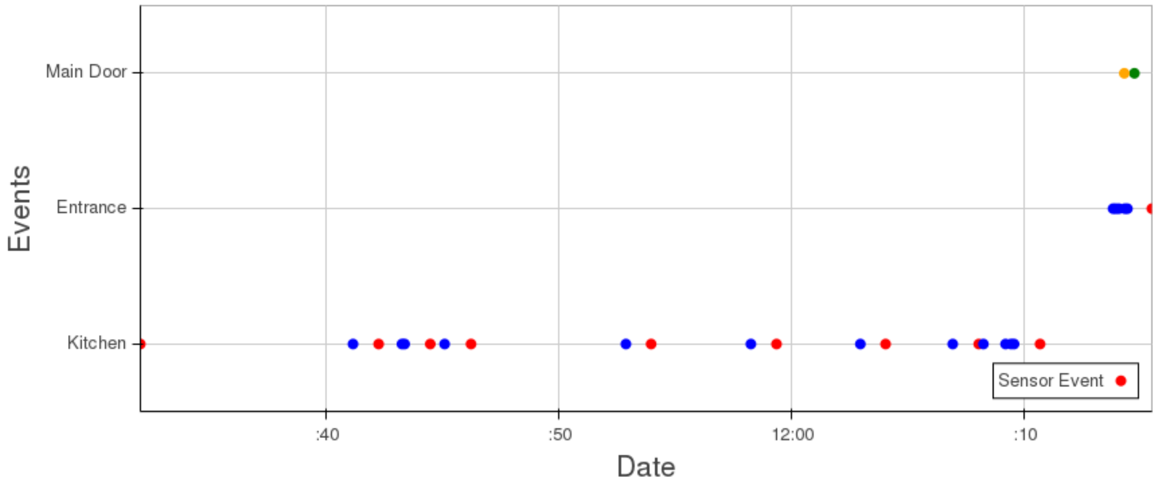


Figure 7.2: Sensor events — Resident taking lunch

In order to highlight the Inaccuracies and Imprecisions that may occur in the activity recognition, I have selected four cases (Figures 7.3-7.6).

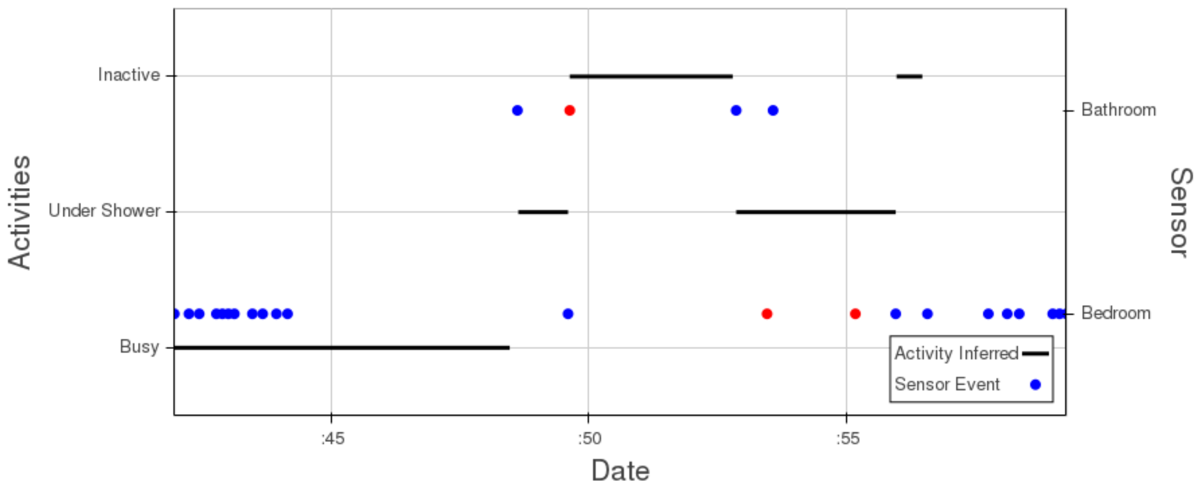


Figure 7.3: Incorrect inference: Reasoner trying to be too Precise

Looking back at the causes of failures, several use cases emerge:

Case 1. Too Precise reasoning engine (Figure 7.3)

By inferring more fine-grained activities, the reasoner diverges from the real activity (which we want to detect). The reasoner infers *"Under Shower"*, whereas the resident is in toilet — based on observations. If the reasoner had inferred *"Is in the Bathroom"*, it would have been less Precise, but it would already have been helpful to reassure the

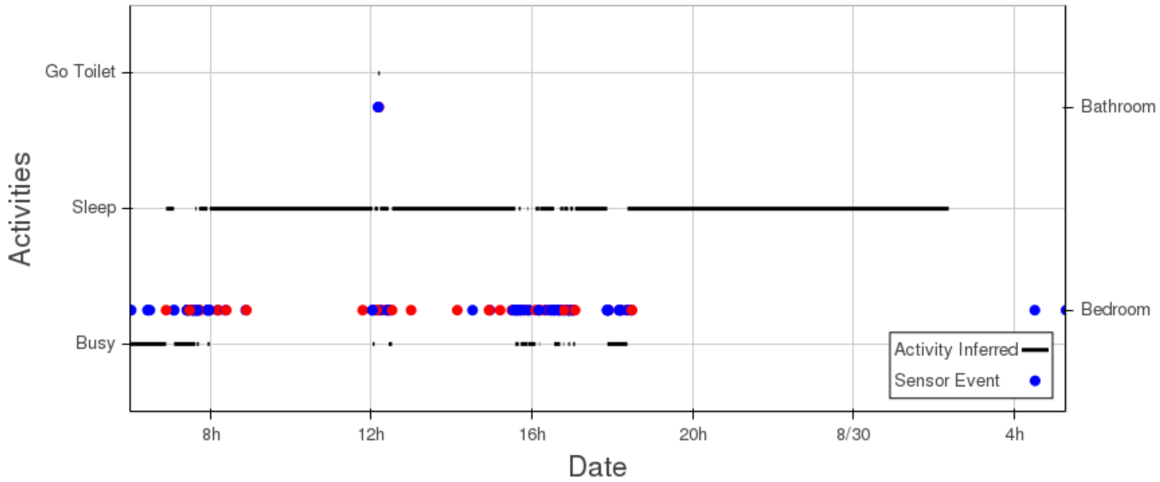


Figure 7.4: Incorrect inference: Faulty rule

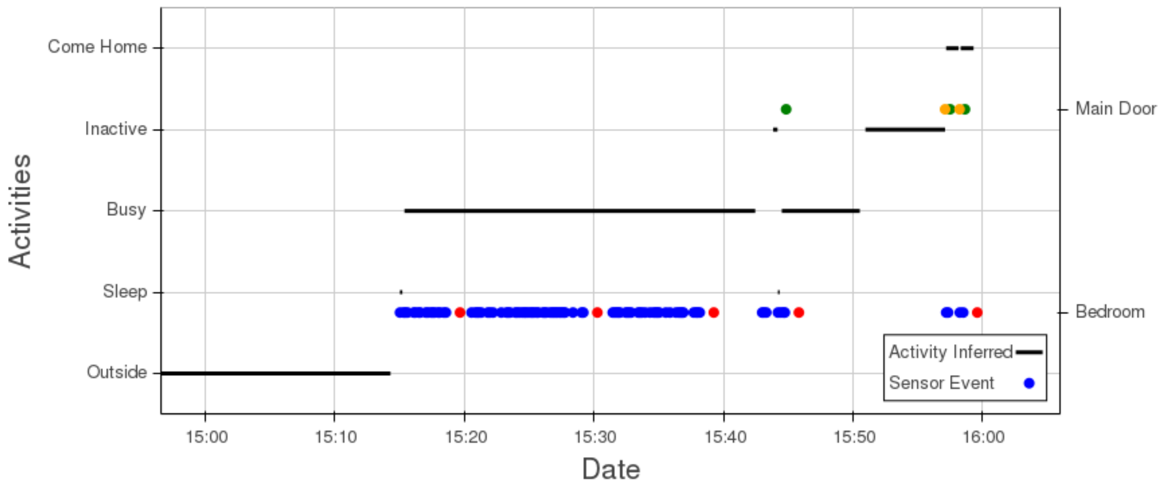


Figure 7.5: Incorrect inference: Reasoner not being Precise enough

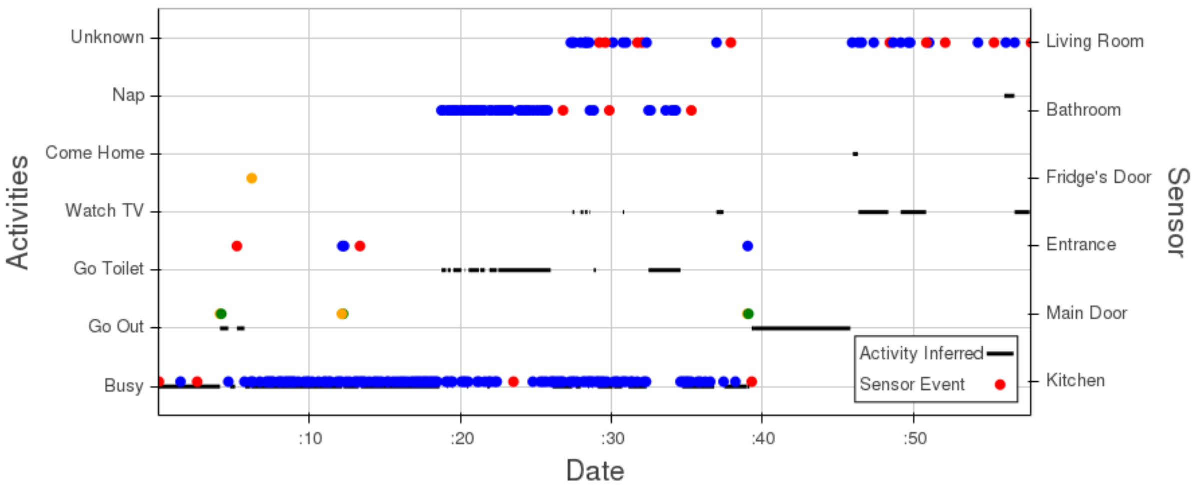


Figure 7.6: Incorrect inference: Reasoner facing confusing events

caregiver, and the system would have been considered more Accurate.

Case 2. Faulty rule (Figure 7.4)

If the reasoner's inference differs from the real activity, while the sensor events are consistent and meaningful, we can suppose that the failure is inherent to the reasoner. In that case, the reasoning rule should be fixed, before any concern on Accuracy and Precision. This situation can be observed in Figure 7.4, where the sensor events clearly show that the resident is active in the bedroom, but the reasoner infers that he is sleeping.

Case 3. Not being Precise enough (Figure 7.5)

Activities such as "Busy" are extremely sparse-grained (i.e., Imprecise). The inferred activity in Figure 7.5 might reassure the caregiver that the resident is not at risk, but this result remains too Imprecise for the system to trigger useful services. In Figure 7.5, we observe multiple sensor events from 15:15 to 15:45, and the reasoner's inference can only infer "Is Busy." In this case, additional sensors should be considered, because no more Precise activity can be inferred from the existing sensor events (even from a human perspective). On the contrary, if the sensor events are meaningful enough to observe more fine-grained activities, the reasoning engine should be adapted to integrate these new activities.

Case 4. Sensor data are confusing (Figure 7.6)

Occasionally, sensor events happen to be confusing (e.g., showing multiple events in multiple rooms). Neither a reasoning engine nor a human analyst could infer what activity is being performed. These can be caused by faulty sensor events, or more likely by having several users inside the residence (which is usual in nursing homes, due to nurse visits).

These multiuser situation have been addressed in Chapter 6.

7.3 Measuring Accuracy of a Rule Engine

I discuss in this section the measurement of Accuracy of the deployed solution. Table 7.1 illustrates an example of Accurate/Inaccurate inferences. As an example of Inaccurate inference, the system infers that the end-user "*Watches TV*" at 14:00, whereas we know from direct observations that he actually "*Takes a Nap*."

Date	**Inferred Activity**	**Observed Activity**
10:00	Sleeps	Sleeps
11:00	Has Shower	Has Shower
12:00	Cooks	Cooks
13:00	Eats	Eats
14:00	***Watches TV***	***Takes a Nap***
15:00	Outside	Outside

Table 7.1: Illustrating Accurate/Inaccurate inferences

There is a need to know how Accurate is a reasoning engine, in order to improve the quality of activity recognition. Thus, there is a need to measure the confidence towards the fact that an activity is actually being performed, given that it has been inferred. To obtain this confidence, we first need to enrich our datasets with real observations (we name them *ground-truth*), in order to confront inferred activities with observed ones. The relation between real activities, sensor events, and inferred activities can be summarized in Figure 7.7. Following we discuss gathering ground-truth and measuring Accuracy of activities.

7.3.1 Gathering Ground-Truth

Ground-truth can be gathered using numerous methods. We introduce in this paper four methods. The first two methods are described in this section, whereas the two others are introduced as perspectives in Section 7.7. The proposed four methods are still work in progress. I highlight in this chapter the strengths and weaknesses for each of them.

Method One: Testing in Laboratory Settings

A method to gather ground-truth would be to reproduce the activities in a laboratory setup, by recreating a realistic environment. In this case, the ground-truth is not spontaneously observed, but it is defined *a priori* in scenarios. The environment has to be equipped with sensors to mimic a real environment. When the actor (i.e., a persons who mimics the activities of an elderly people) executes the scenario, sensor data are collected, and a reasoning is performed.

This method is more flexible than observing ground-truth in a real deployment. However, the observations may be flawed, as some activities may be hard to reproduce. We

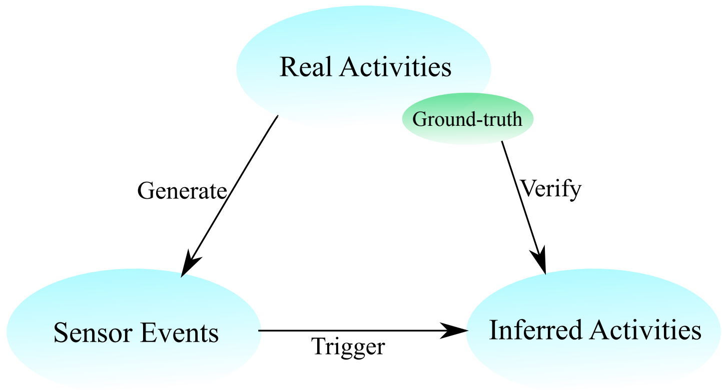


Figure 7.7: Relation between Real Activities, Inferred Activities, and Sensor Events

also probably miss emerging behaviors that had not been expected (e.g., a person could perform her morning routine in an unexpected way because of a sickness). Similarly, in a laboratory experiment, we are less subject to technical issues such as network issues or sensor failures, whose effects on the reasoning should be acknowledged. These laboratory experiments are helpful to punctually compare the execution of an activity with the inference engine, but they does not bring any feedback about the habits of an end-user, neither on frequency of an activity over a day.

Laboratory experiments can be acceptable in the ground-truth acquisition, only if there is no need to know the frequency of an activity, and assuming that the activities are being performed as they would be in a real situation.

Method Two: Acquiring Feedback Through Direct Observation

This method provides the greatest benefits, but it is also the hardest to set up. In this method, a human regularly observes and records what end-users do. The result is a list of punctual observations of activities over a period of time. Through this method, we can be certain that the observations represent real situation. However a bias may be introduced by the sampling periods: the observers are more likely to perform the ground-truth acquisition only at specific hours of the day (e.g., in morning time for nurses), ending up with an heterogeneous density of data, that must be translated later in the measurement process. The ground-truth acquisition is a manual process, and it is not immune to human errors. This manual process is time-consuming, which restricts observers recruiting.

This method may also have logistic difficulties and acceptance issues, particularly in the case of collecting data linked to end-user living independently. I have experienced this situation ourselves in individual houses. A solution would be to ask caregivers to perform acquisition, but it would only bring little benefit, as caregivers would actually influence the environment they observe. In this situation, ground-truth acquisitions would take place in a multiuser situations (i.e., end-user and caregiver), in which the acquired data sensors cannot be considered linked only to the end-user.

The observations are more simple in a nursing home, as many end-users keep the door open most of the time, and it is easy to see what they are doing without influencing the environment.

Another approach to overcome the limitations of the direct observation by human would be using cameras. This method is rich in data, however we did not consider it due to acceptability and privacy concerns. In our experiments, we have prototyped a simple smartphone and tablet application (Figure 7.8), where observers can easily input the activity being performed, the end-user's location, and even specify a custom time if an action happened in the past. The gathered data is sent directly to our server and saved in database, where we are ready to analyze it.

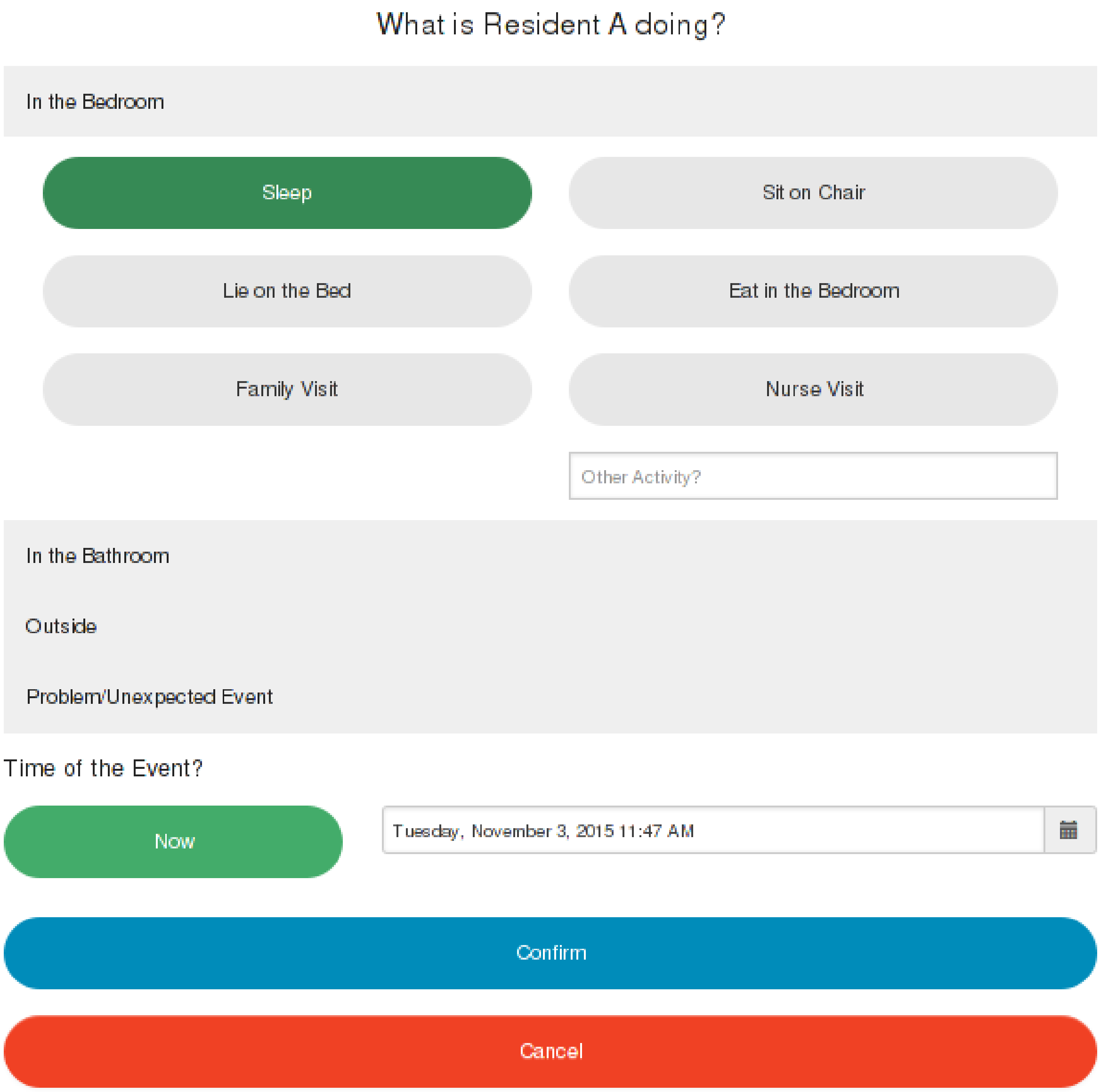


Figure 7.8: Smartphone application to gather ground-truth

7.3.2 Measuring Accuracy of Activities

One goal of my research is to give a confidence value to an inferred activity based on our ground-truth observations of real activities. More precisely, I am interested in a metric based on the probability $P(A = a|I = a)$ that an activity a is being actually performed (A), given that it has been inferred (I). In other words: "is the person really doing a, when a reasoner says a?" I define the metric as follows:

$$
\begin{aligned}
P(A = a|I = i) &= \frac{P(A = a \cap I = i)}{P(I = i)} \\
&= \frac{|a \cap i|}{|A \cap I| P(I = i)} \\
&= \frac{|a \cap i| \sum_{X \in I} duration(X)}{|A \cap I| \sum_{x \in i} duration(x)}
\end{aligned}
\tag{7.1}
$$

where:

$|A \cap I|$ is the number of observations made while inferring an activity

$|a \cap i|$ is the number of occurrences when i is inferred and a is observed

$\sum_{x \in i} duration(x)$ is the total duration when i is inferred

$\sum_{X \in I} duration(X)$ is the total duration covered by our inferences

In the case when the activity has never been observed or never been inferred, I set Accuracy value of this activity as 0.

7.3.3 Discussion

I have introduced in this section a metric to measure Accuracy of a reasoning engine, based on observations. We also have described two methods to gather observations, whether from laboratory experiments, or from direct observations on real deployments. Laboratory experiment is more flexible and gives more control on the activities to test, whereas real observations return more reliable observations but are more tedious. I have used both methods in my research, and I can confirm that using a more flexible method tends to generate less reliable observations. There is a trade-off between the flexibility and reliability of the observation methods based on the need of the studies. Following I discuss how to use Accuracy to improve the decision-making process.

7.4 Improving the Decision-Making Process By Introducing A Score Of Quality

I believe that the Accuracy metric can be helpful to evaluate the quality of a reasoning engine. By coupling Accuracy with Precision, and by introducing hierarchical models, I propose a systematic method for the reasoner to conclude with more effective inferences.

7.4.1 Introducing Precision to the Reasoner

As shown in the example of Figure 7.3, the reasoner infers “Takes Shower.” However, our ground-truth shows a faulty inference, as the end-user “Goes to Toilet.” As for the example of Figure 7.5, the reasoning engine can infer “Is Busy” only, which is not an adequate inference for a service delivery. These two cases are caused by an irrelevant Precision. In the example of Figure 7.3, the reasoner tends to be too Precise, and leads to an Inaccurate inference; “Is in the Bathroom” is a slightly less Precise activity than “Takes Shower,” and our reasoner would be able to infer it more Accurately.

Generally, a human observer intuitively adjusts his conclusions to the expected level of Precision. He also measures the risks of being Inaccurate when he makes Precise conclusions. My approach is to apply the same process for decision-making in activity recognition, so that the system will minimize the risk of Inaccuracy and maximize Precision. Thus, I represent Precision in the model as a value ranging from “1” to “10,” determined arbitrarily by an expert. “1” would be an extremely Imprecise activity and “10” would be an extremely Precise one. I argue that introducing Precision into the model as an arbitrary value is acceptable, because an acceptable Precision is a non-functional requirement of a system, which is by nature arbitrary.

7.4.2 Introducing Hierarchical Activities

Looking at Figure 7.3, we can intuitively perceive that “Takes Shower” and “Goes to Toilet” can both be generalized as “Is in the Bathroom.” In other words, “Is in the Bathroom” is a parent activity of “Takes Shower” and “Goes to Toilet.” More generally, activities exist at different granularities, and can be modeled as hierarchies. With a hierarchical model, several activities are inferred at a given time, and a person can be both “Going To Toilet” and “In the Bathroom.” When an activity occurs, all of its generalizations occur, recursively. Similarly, we can affirm that when an activity occurs, some of its specializations may occur too. An example of a hierarchical model is presented in Figure 7.10, and the generalization inference goes from right to left.

Hierarchical models are richer than linear models, and the ability to infer activities on several layers has powerful applications when combined with Precision and Accuracy. In a hierarchical model, an activity always has a higher Precision than its parent. On the other hand, an activity always has a lower Accuracy than its parent: if an activity is Accurate, its parent will always be Accurate, but if an activity is not Accurate, its parent will sometimes be Accurate. Following this logic, I created a formal process to perform Accurate inferences while being as Precise as possible. The reasoning process is similar to a tree exploration, where the system starts by reasoning on the least-Precise activities (i.e., the activities with no parent) to its specializations (more Precise). In this path (parent → kid), the reasoning checks whether the context can be valid or not (for of each specialization). This process can be executed recursively across specializations, until the inference of a chain of activities, from the least Precise to the most Precise (Figure 7.9). I expect that with this process, a system converges towards the activity that has the best balance between Precision and Accuracy.

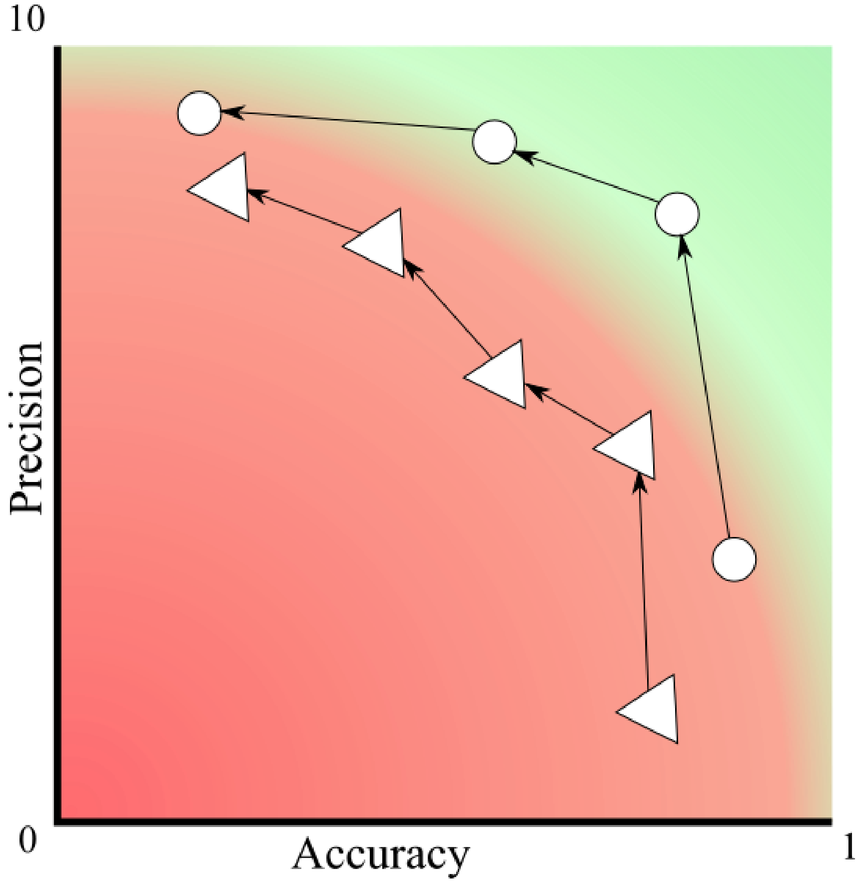


Figure 7.9: Illustrating the relation between Accuracy and Precision

7.4.3 Measuring Performance of the Reasoning Engine

I propose to formalize the process of inferring activities with the right level of Precision. Therefore, I introduce a Score of quality of activity recognition that enables the reasoner to converge towards the activity having the best balance of Accuracy vs. Precision. The proposed process starts by giving each activity a Score, that is based on its Accuracy and Precision. Then, the decision-making engine selects the activity with the best Score. The system uses a weighted geometric mean of Accuracy and Precision (Equation 7.2) to

- Activities
  - Kitchen Activity
    - Eat Meal
    - Cook Meal
  - Activity in Bathroom
    - Go Toilet
    - Hygiene
  - Living Room Activity
    - Read Book
    - Watch TV
  - Bedroom Activity
  - Sleep
  - Outside
  - Come Home

Figure 7.10: Hierarchical activities in the UbiSMART reasoner

measure Score. The advantage of a geometric mean is that a marginally low value of either Precision or Accuracy has a dramatic impact on the resulting Score, and an Accuracy of 0% will result in a Score of 0.

$$Score(a) = \left(accuracy\,(a)^{A} \times (0.1 \times precision\,(a))^{P}\right)^{\frac{1}{A \times P}} \tag{7.2}$$

Where:

A is the weight given to Accuracy

P is the weight given to Precision

The 0.1 factor is used to normalize Precision as a $[0, 1]$ value

A and P are to be defined by experts. However, there is no objective criteria to set them. From my empirical experience, I choose to give more importance to Accuracy than to Precision, and to set $A = 3$ and $P = 1$. The goal is to have the most useful sets for the end-user experience. One may also consider running reasoners with various values of A and B, and ask end-users which reasoner generates the most useful conclusions.

7.4.4 Discussion

I have introduced in this section a Score that represents the quality of a reasoner to infer an activity. I also have discussed the use of our two metrics (i.e., *Accuracy* and *Precision*) in Score, in order to enable reasoning engines to converge towards the most rational conclusion among possible activities in a hierarchical chain. Score also can be used to choose between two or more inferred activities at a time t (e.g., conflicting reasoning).

The Score calculation method presented above is a simple weighted geometric average calculation. Several approaches can improve the computing of this Score. Score can also be improved by enriching the concept of Precision, for example by dividing Precision into several criteria (e.g., criticality, interest).

The method I propose in this paper does not require to redefine reasoning engines. The approach is reasoner-agnostic, as long as the used reasoner includes Precision and Accuracy. In addition, preferably reasoners should define activities in a hierarchy. The proposed method is validated in Chapter 9.

7.5 Integration

7.5.1 Update the UbiSMART Reasoner

The UbiSMART reasoning engine uses ontological models [30, 31], and infers through the Euler Rule Engine [34]. The original version does not include the hierarchical activities approach. It embedded an arbitrary Score in the reasoning engine. For the purpose of this chapter, I have made several improvements to the UbiSMART reasoning engine that can be summarized as follows:

Introducing the Hierarchical Activities Method
Since UbiSMART is ontology-based, I have formally defined the concepts of *generalization* and *specialization* into the ontology (Section 7.4.2). I have also adapted the rule engine, so that it infers on parent activities first, and iterates recursively to the children activities.

Introducing Activities Scores
The original UbiSMART reasoner applied arbitrary Scores to activities. I change that by calculating Scores of each activity based on Equation 7.2, given Accuracy measured from our ground-truth. Using this Score, the reasoner can conclude on the most reasonable activity (i.e., the activity that has the maximal Score among all the inferred activities).

7.5.2 Run The UbiSMART Reasoner

In order to validate this method, I use the same datasets (Section 9.4.2) to run the UbiSMART's reasoner, before and after applying the proposed method. The reasoning engine can be executed in two different ways.

1. In production mode, we execute the reasoner in order to infer a single activity at a given time. In this case, Accuracy has to be predefined. This is the mode used for the UbiSMART system to deliver services.
2. In calibration mode, we execute the reasoner in order to measure Accuracy of all activities in hierarchical chain. We expect the reasoner to return all possible valid inferred activities, without making a conclusion. The result of a calibration mode should are used as an input for the production mode. The calibration mode is run

only at the first phases of deployment. Once the reasoner has converged towards stationary Accuracy values, there is no need to run calibration mode anymore.

7.6 Discussion

I have introduced in this chapter a method for improving Activity Recognition decision-making process in developing AAL solutions. This research has been motivated by the feedback we had from our real deployment in a nursing home and three individual houses. The observations from this real deployment brought my attention to the faulty results of our reasoning engine, and stressed the need for a systematic method to evaluate reasoning engines in order to improve the activity recognition process.

I argue that an efficient systematic method to evaluate reasoning engines has to include ground-truth from deployment environment (e.g., elderly people house). Therefore, I proposed in this paper a method that includes observing ground-truth as an input for measuring Accuracy of a reasoning engine. I also introduce for the first time Precision and Score of the quality of a reasoning engine. The Score is derived from Accuracy and Precision. This method effectively leads to conclude on the most reasonable activity (i.e., the activity with the best balance between Precision and Accuracy).

The method introduced in this paper is dependent on a *ground-truth* knowledge of the activities performed, in order to measure Accuracy or the activity recognition engine. I believe that it could be improved either by finding alternative sources of ground-truth, or by involving the caregiver in the process of observing ground-truth in a deployment.

7.7 Perspectives

One of the difficulties that I faced in real deployment is the logistic difficulty to observe the ground-truth, that is the keystone of Accuracy measurement. The real deployment generates limited quantity of data, which may affect the results of the reasoning process. Therefore, other alternatives to enrich and optimize the ground-truth may be beneficial. Following, I discuss a number of these alternatives.

7.7.1 Detecting Faulty Rules

As the value of Precision is static, and the Accuracy is calculated offline, the Score of an activity is never going to change on runtime. In other words, activities with a low Accuracy and a low Precision will always have a low Score.

Imprecise activities are not considered particularly problematic, as they might be parents of more Precise activities. However, Inaccurate activities (regardless whether they are Precise or not) are problematic, and might be caused by a poorly defined rule.

The most problematic case is *faulty chains*. These are hierarchies where none of the activities are acceptable, because they are either too Imprecise or Inaccurate. In Figure 7.9, it is represented by the triangles, where every points are in the red zone. Fortunately, it should be possible to automatically detect faulty chains, and notify the expert.

Generally speaking, a higher Precision comes along with a lower Accuracy. But in some extreme cases, like inferring from *Cooks* to *Cooks Pasta*, Section 8.3.1), we would measure an extremely strong drop of Accuracy, for rather small gain in Precision. These are the kind of activities we want the system to recognize and notify to the expert.

To do so, the system would measure the difference of Accuracy between an activity and its parent, as well as the difference of Precision. Equation 7.3 measures at what speed the Accuracy decreases per precision unit. I name this value $Q(a)$, the isolated quality of an activity a. The highest $Q(a)$ is, the most poorly the activity a has been defined. It is worth mentioning that activities with no parent have no value of Q.

$$Q(a) = \frac{Accuracy(parent(a)) - Accuracy(a)}{Precision(a) - Precision(parent(a))} \tag{7.3}$$

By calculating this value for every parent-child activity couple, the system can detect where the accuracy decreases too fast. I suggest using Theory of Constraint [324] to rank activities in decreasing order, starting with the highest value of Q. The activities with the highest Q are the ones that should be investigated first by the expert.

7.7.2 Enriching Ground-Truth By Using Simulators

A first alternative approach to gather feedback is an extension of laboratory experiments. Rather than simulating environment in a lab, we can reproduce the experiment over a computer simulator. The approach of using simulators brings a great flexibility. It can generate a large amount of data in a short time. Through stochastic simulations, we also could introduce variations between the iterations, to gather richer datasets. Moreover, whereas experiments in living lab reproduce isolated activities, a computer simulator could generate a consistent sequence of activities over several virtual days. Still, there is a gap between a data from a simulator and a real situation. The simulator generated data could be faulty and misleading.

I have developed a simulator during my thesis work (Chapter 5.5), to generate realistic sensor events from scenarios. Based on this application, I conclude that simulators are particularly useful for fast prototyping. Indeed, simulators enable to define a scenario that represents a supposed sequence of actions in a defined activity. This sequence can generate sensor events in order to check whether the activity is recognized correctly in an ideal situation.

A weak point of simulators is that there is no way to guarantee that they are realistic. Simulators can not include all possible scenarios and problems of real situations. Yet, there might be a way to guarantee that a simulation is realistic. Assuming that the simulator's environment is based on a real deployment (e.g., same rooms, same sensors), one can measure the similarity between the sensor data from the simulator and the ones observed in the real deployment. One could them estimate how distant the generated events are from reality, by quantifying the similarity between the two datasets of sensor events, and give a confidence score to the simulator accordingly.

7.7.3 Enriching Ground-Truth By Relying On User Habits

A second alternative approach to gather feedback can be to ask end-users (i.e., the elderly people, or their caregivers on behalf of the elderly people) about their habits. This approach would enable to measure the similarity between an inferred activity and that occurred in real life.

An obvious problem with this method is the comparison between inferred activities and habits, and not a comparison between inferred activities and an end-user's activities in a specific day. I believe that a solution can be to aggregate sensor data over several days, in order to obtain an overview of habits. If the sensor data at a specific time are close to an aggregated one, the end-user people is certainly performing his habit, and it can be compared with the inferred activities.

7.7.4 Optimizing the Ground-Truth Acquisition By Involving Caregivers

Elderly people are frequently in interaction with their caregivers (e.g., doctors, nurses). Caregivers can hold rich information for the ground-truth. They can make ground-truth observations more consistently in a real deployment (as they are in continuous interaction with the end-users). A solution would be to involve these caregivers, in order to gather observations. I have investigated this approach and the result were encouraging. Still, automating the process in order to gather data related to a specific time-spot could improve the quality of the reasoning. I have considered on a tool that automatically sends requests to caregivers to collect the needed observations. I have identified three possible policies for the frequency of the requests:

Random requests
: This first policy would consist of sending request randomly. The benefit of this policy is that random requests generate homogeneous observations, whereas more advanced request policies tend to generate heterogeneous observations. Homogeneous observations are more adequate for statistical purpose. Heterogeneous observations can be more insightful for Accuracy measurement process, assuming that density of the observations is higher in defined situations.

Request for an inferred activity
: As an activity becomes more observed, Accuracy measurements converges to a stationary value. Thus, this policy focuses on requesting ground-truth on the inferred activity, in order to better estimate Accuracy.

Request for an observed sequence of sensor events
By using sequential clustering, it is possible to detect frequent sequential patterns of sensor events [261]. Arguably, these frequent patterns are representative of an activity performed multiple times. We could therefore request feedback when an unknown pattern is observed, in order to maximize the entropy of the ground-truth.

7.7.5 Enriching Ground-Truth From Services

I believe that taking full advantage of the context-awareness and use the context as an input in our reasoning engine process can improve Accuracy measurement. As an ambitious perspective, I would consider the option of redesigning the services we are deploying in real residences in order to collect context that is useful for the process. In addition, I believe that embedding metrics into the delivered services would confirm — or deny — that a user is performing or not an inferred activity. That method would not require any effort from users, and would be more consistent with the paradigm of Pervasive Computing, where the system must adapt to users, rather than asking for explicit feedback from users.

The successful man will profit from his mistakes and try again in a different way.

Dale Carnegie

8 Detecting Inconsistencies in Rule-Based Reasoning

8.1 Introduction

In this chapter, I propose a set of methods designed to detect inconsistencies in rule-based reasoning engines. The purpose of these methods is to facilitate the creation of an information system for AAL using ontological model and rule-based reasoning. To do so, I rely on an abstraction layer on top of the models and rules, which enables dynamic definition of an environment (named *houses factory*) and creation of reasoning rules (named *rules factory*). The *houses factory* and *rules factory* features have been developed as core components of Knot (Chapter 5).

Ontological models are popular in AAL application, as they have the flexibility to model complex situations. Rule-based reasoning engines are often used in complement to ontological models, to perform activity recognition.

Ontological models and rules-based reasoning engines have proven their value in final environments. Yet, I have experienced that designing such systems was uneasy, and error-prone. Little refactoring take place in models and rules, and experts often find themselves with redundancy between the various rules' content. Practically speaking, the usual way to design such applications is by editing graphs, or plain-text ontological files. Either case, experts have to deal with the raw complexity of ontological models, whereas they only use a finite subset of the features.

In the contribution presented in this chapter, I hope to facilitate the expert's work, reducing the number of errors in AAL information system, and encouraging faster prototyping iterations.

8.2 Dynamic Environment Definition

In this section, I describe a solution for helping the expert to define an environment and a set of rules. Similar solutions have been proposed in the literature as early as 2005, where Helal et al. [65] introduced a *context-builder*. But the solution I propose is not only a content generator, it also builds an API-type access on top of the ontology, so that external programs can interact with it easily. The API-type access is a prerequisite for the algorithms described in Section 8.3.

Traditional ontology-based AAL applications interact directly with the ontology, to define the environment. I believe that core ontology should be decoupled from the high-level models and rules. The AAL expert should not be concerned by the underlying complexity of ontologies, and should solely focus on his AAL toolkit.

In response, I propose a tool to define the environment dynamically. Using this tool, the expert can programmatically define a house and insert rooms, objects, activity recognition rules, residents, etc. to the ontology, suing well-defined methods. The two primary features I have developed are the *houses factory* and the *rules factory*. The underlying idea behind both is to refactor the content of our ontology down to a finite (yet extensible) set of templates, which will be instanciated by the experts.

In these features, I favor a *convention over configuration* approach, where the expert is not captive of the features, and can still manually edit the ontology if required.

8.2.1 Houses Factory

The purpose of the houses factory is to provide a high-level interface for defining an AAL environment (e.g., a smart-home).

Generally, an AAL environment is composed of one house, itself composed of rooms, doors, and objects. These could be equipped with sensors. The house is inhabited by a resident, who may receive visit from other persons. I have defined templates for each component mentioned in this use-case, as reported in Table 8.1. Other types of components should be included later on, if the use-case broadens.

Model	**Arguments**
House	
Room	Room type
Door	From room To room
Item	Object type In room
Sensor	Sensor type Attached to
Person	

Table 8.1: Existing components of the house factory

Templates should be defined programmatically, and maintain an ontological model in the back-end. To do so, I suggest the use of a ORM. ORMs are traditionally designed to provide an object-type Application Programming Interface (API) access to a back-end database. They are often used in the Model layer of Model-View-Controller (MVC) architectures to create an abstraction layer over the database. In Knot, I propose to use an ontological ORM (Figure 8.1), so that RDF classes are defined as object-oriented programming classes, RDF properties are defined as object-oriented class attributes, and RDF individuals are instanciated as programming objects. Through the use of an ontological ORM, one can use Create, Read, Update, Delete (CRUD) interactions to easily manipulate the ontology. The ontological ORM should also integrate inheritance, using RDF's "*subClass*" property.

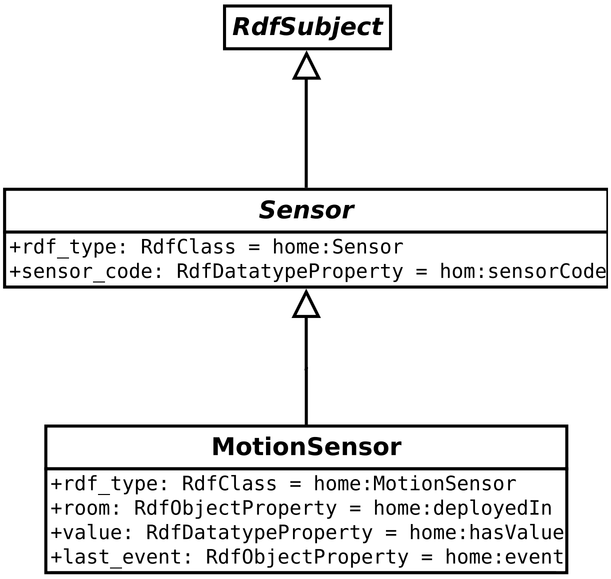


Figure 8.1: Defining sensor models with the ORM

If the developer wants to use an object of the environment that is not part of the *houses factory* yet, he should define a new *class* for that object.

To instanciate a new environment, we propose a JavaScript Object Notation (JSON) interface (Listing 8.1). A JSON-defined environment would be entirely declarative, and require only a minimal technical knowledge from the expert.

```
[{ "house_name": "House 1",
    "rooms": [
        {"name": "Bedroom",
         "type": "bedroom"},
        {"name": "Living Room",
         "type": "livingRoom"}],
    "doors": [
        {"doorFrom": "room:Bedroom",
         "doorTo": "room:Living Room"}],
    "items": [
        {"type": "bed",
         "name": "Main bed",
         "locatedIn": "room:Bedroom"}],
    "sensors": [{
        "code": "A1",
        "type": "Motion Sensor",
        "attachedTo": "room:Bedroom" }],
    "persons": [{"name": "John Doe",
        "type": "Resident" }]
}]
```

Listing 8.1: Defining a house using the Houses Factory

8.2.2 Rules Factory

In an ontological rule-based reasoning, rules are defined using a IF-THEN model. If every conditions are validated on the left-hand side of the rule, the assertions written in the right-hand side are inferred, and added to the knowledge base. After every new external input, the new knowledge triggers new rules, which infers new knowledge, which itself triggers new rules, until reaching a stable state where no additional knowledge can be inferred.

When applied to AAL, one can again define a finite number of types of condition that can trigger an activity rule (Table 8.2). Each of these types of condition is implemented as a condition template (Table 8.2), which are mapped to RDF assertions. The developer's work is then to create a rule that associates an activity with a set of conditions, based on the available condition templates. In case the expert needs to add a new condition that does not match any existing template, it is suggested that he creates a new template.

Similarly to the *houses factory*, the instanciation is defined by using a declarative JSON format (Listing 8.2), so that the expert only needs a limited technical knowledge.

Condition	**Related Variable**
Hour of the Day	$\emptyset$
Coming From	room
Detected In	room
Previous Activity	previous activity
Using Item	item
Involve Presence	person
Max Duration	$\emptyset$
Cooldown	$\emptyset$

Table 8.2: Existing condition patterns in the rules factory

```
[{
    "name": "Sleep",
    "conditions": {
        "hour_of_the_day": [20, 10],
        "detected_in": {"room": "room:Bedroom",
                        "max_motion": 1},
        "max_duration": "10h",
        "cooldown": "3min"
    }
}]
```

Listing 8.2: Defining a rule using the Rules Factory

8.3 Detecting Problematic Reasoning Rules

Rule-based reasoning engines are powerful in AAL, but they rely heavily on the expert's work, and therefore are vulnerable to human errors. In this section, I introduce two methods to assist the AAL expert by detecting inconsistencies in his model:

- Detect rules whose conditions cannot be satisfied, and suggest new sensors accordingly;
- Detect rules conflicting with each other.

These methods are semi-autonomous; they automatically detect issues in the rules definition and notify the expert about these issue. At the end of the process, the expert is the one who takes action.

8.3.1 Suggesting new sensors

Thanks to the rules factory (Section 8.2.2), rules are created upon on a finite set of condition patterns. It is possible to generate the list of sensors required for each condition of a rule. Table 8.3 lists every condition patterns implemented in the rule factory at the moment, and the required sensors for each of these conditions. Given a hierarchical model for activities (as introduced in Section 7.4.2 from Chapter 7), if a parent activity requires sensors that are missing, none of the children activities can ever be inferred.

Context	**Related Variable**	**Sensors Required**
Hour of the Day	*None*	*None*
Coming From	room	Motion Sensor(room)
Detected In	room	Motion Sensor(room)
Previous Activity	previous activity	Requirements(previous activity)
Using Item	item	Accelerometer(item), etc.
Involve Presence	person	*To be determined*

Table 8.3: Requirements for each rule template

Therefore, the method I propose can predict that activities which require missing sensors will never be inferred, and possibly suggest the sensors that are missing. For instance, the activity "In the kitchen" would require *Detected In Kitchen*, which requires a motion sensor in the kitchen. Additionally, "Cooking" would require *Using Item Stove*, that could require an electricity meter. "Cooking pasta" would also require *Using Item Pasta*, that could even require a RFID sensor on the pasta box. One can observe that extremely precise activities tend to require extremely precise sensors.

I propose the Algorithm 1 to detect and suggest missing sensors.

Algorithm 1 Detecting missing sensors in a rule

```
Inputs:The rule to check, The list of sensors
Output:The unmatched requirements, empty if everything is correct
function FIND_MISSING_SENSORS(rule, sensors)
    missing ← ∅
                        ▷ We check every requirements of every conditions of the rule:
    for all condition ∈ rule.getConditions() do
        for all requirement ∈ condition.getRequirements() do
            matching ← False
            for all sensor ∈ sensors do
                if requirement.matchedBy(sensor) then
                    matching ← True    ▷ There exists a sensor that fits the requirement
                    break
                end if
            end for
            if not matching then                ▷ If no adequate sensor was found:
                missing.append(requirement)
            end if
        end for
    end for
    return missing
end function
```

8.3.2 Finding Conflicts Between Rules

Another inconsistency that may happen is when two rules are overlapping under certain conditions (i.e., both rules may be recognized recognized at the same time, with no possibility of choosing). This use case has been handled in Chapter 7, by tagging each activity with a score. Yet, the expert could be interested to observe some of these overlapping situations, as some of these may be symptomatic of a poorly defined rule.

Algorithm 2 shows the algorithm used to detect conflicting rules. For each condition pattern, I define a function `intersection(self, other))` ($\cap$), to determine the intersection between two conditions. If dealing with a hierarchical model, as introduced in Chapter 7, conflicts between rules of the same hierarchy will be ignored. As an example, the intersection between [22:00 — 02:00] and [01:00 — 10:00] is [01:00 — 02:00], and the interval between "Detected in Bedroom" and "Detected in Bathroom" is $\emptyset$ (Algorithm 3).

Algorithm 3 Implementing the *intersection* method for *location* condition

```
function INTERSECTION_DETECTED_IN(self, other)
    return self.room() ≠ other.room()
end function
```

Algorithm 2 Detecting conflicts between rules

```
Inputs:Two rules.
Output:The list of overlapping conditions, empty if there is no conflict.
function FIND_CONFLICTS(rule1, rule2)
    conflict ← ∅
    conditions_templates ← GET_ALL_CONDITION_TEMPLATES(rule1, rule2)
    for all template ∈ conditions_templates do
        condition1 ← rule1.GET_CONDITION_BY_TEMPLATE(template)
        condition2 ← rule2.GET_CONDITION_BY_TEMPLATE(template)
        if INTERSECTION(condition1, condition2).EMPTY then
            ▷ If there is one condition without overlapping, the rules are not conflicting
            return ∅
        end if
        conflict.APPEND(template)
    end for
    return conflict
end function

function GET_CONDITION_BY_TEMPLATE(rule, template)
    for all condition ∈ rule.CONDITIONS do
        if condition.TEMPLATE = template then
            return condition
        end if
    end for
                    ▷ If the rule does not specifies this template, we create a condition
default ∈ template, so that x ∩ default = x.
    return template.MAKE_DEFAULT_CONDITION()
end function

Inputs:Every rules
Output:A list of tuples < rule1, rule2, conflict >
function FIND_ALL_CONFLICTS(rules)
    all_conflicts ← ∅
                                        ▷ We iterate over each combination of rules:
    for i ← 0; i ≤ LENGTH(rules); i++ do
        other_rules ← ∅
        for j ← i + 1; j ≤ LENGTH(rules); j + + do
            conflict ← FIND_CONFLICTS(rules[i], rules[j])
            if not (COMMON_HIERARCHY(rules[i], rules[j]) or conflict.EMPTY) then
                all_conflicts.APPEND([rules[i], rules[j], conflict])
            end if
        end for
    end for
    return all_conflicts
end function
```

8.4 Implementation

The contributions of this chapter have been implemented as part of Knot, hence using the Python language.

The interactions with the ontology are performed using the RDFLib library. In order to implement the ORM described with the *houses factory*, I have used the RDFAlchemy library[1], which defines itself as an ORM for RDF.

8.5 Conclusion

The two methods presented in Section 8.3 offer practical and useful insights for the expert to detect issues on the reasoner, so he can take any necessary action to improve it. More generally, I believe that numerous similar applications could be developed, leveraging the contributions of this thesis (e.g., Precision and Accuracy, hierarchical activities) in order to assist the expert in a process of continuous improvement for activity recognition. The major prerequisite for developing such methods is to have refactored the creation of a model and rules into a finite number of templates, as described in Section 8.2.

This contribution is validated in Section 9.5 from Chapter 9.

8.6 Perspectives

The contributions described in this chapter are proof-of-concept. I foresee three possibilities for improving them and making them more industry ready.

First, it has been mentioned in Chapter 5 that Knot is still a work-in-progress. One of the perspective I listed was to propose a modular architecture to Knot, that would integrate a system of plug-ins. Following this perspective, templates from the *houses factory* and the *rules factory* should be defined as plugins as well.

A JSON interface for the *factories* is developer-friendly, but might be uneasy to understand for a non-technical user. I believe that a Graphical User Interface (GUI) should be proposed, in order to guide the expert user more effectively.

As a final perspective, I would like to note that there is no reason why this contribution should be bounded to AAL. The proposed methods could be extended to any ontological-based reasoning, as long as a *factory* is created for refactoring the environment.

[1] `http://openvest.com/trac/wiki/RDFAlchemy`

Part V

Validation

Building the right product requires systematically and relentlessly testing that vision to discover which elements of it are brilliant, and which are crazy.

Eric Ries

9 Validation

In this chapter, I validate the contributions presented in Chapters 4, 5, 6 and 7. Section 9.1 validates the UbiSMART framework, with a focus on usage and data acquired (based on Chapter 4). Section 9.2 offers a functional validation of the Knot prototyping framework, through an integrated example (based on Chapter 5). Section 9.3 validates the multiuser detection (based on Chapter 6). Section 9.4 validates the method proposed in Chapter 7 to make activity recognition engines more accurate. Section 9.5 validates the contributions of Chapter 8 for detecting inconsistencies in rule-based reasoning engines.

9.1 Validating UbiSMART As A framework For Ambient Assisted Living

9.1.1 Deployment

My colleagues an I have collected a year of data over a deployment in one nursing home and one individual residence in Normandie, France. Two other houses where deployed, but the data was not significant, and are not included in this section.

In the nursing home, each resident lives in his own room. We have deployed 5 rooms over a corridor. Each room is composed of a bedroom and a bathroom. We have deployed one motion sensor in the bedroom, another motion sensor in the bathroom, and a door sensor on the entrance door. A bed sensor has been prototyped in one room for a period of 6 months, telling us whether the resident was on the bed or not. Sadly, several residents passed away during the period of deployment. Their room has then been occupied by other persons. This leads to a total of 9 residents in the nursing home who have been equipped with our system.

As for the individual residence, the person was living in her own apartment, in a small city. She deceased in January 2016. The floorplan of her apartment can be seen in Figure 9.1, as well as the sensors we have deployed. She had her full cognitive capacities, and although she was frail physically, she remained autonomous and sometimes went out to meet her neighbour. She received regular visits from her family and from professional caregivers.

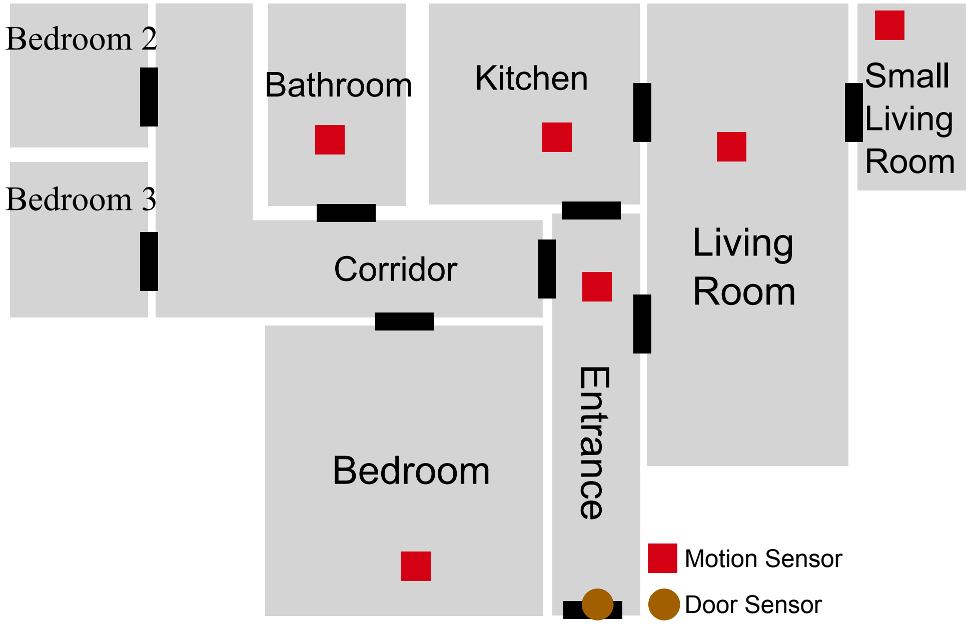


Figure 9.1: Topology of the first house deployed

For each elderly people, we save into database the sensor events, and the activities being inferred by our reasoning engine. Table 9.1 summarizes the amount of data we have for each resident.

Residence	Room	Start of deployment	End of deployment	Events	Activities inferred
Nursing Home	Resident 1	23 Sep 2014	14 Mar 2015	29,194	6,491
	Resident 2	15 Mar 2015	26 Aug 2015	34,680	10,101
	Resident 3	28 Aug 2015	Ongoing	5,210	1,243
	Resident 4	23 Sep 2014	06 Aug 2015	63,715	12,897
	Resident 5	07 Aug 2015	Ongoing	2,891	428
	Resident 6	23 Sep 2014	Ongoing	85,231	15,946
	Resident 7	23 Sep 2014	Ongoing	44,102	10,332
	Resident 8	25 Sep 2014	03 Jun 2015	10,576	2,673
	Resident 9	04 Jun 2015	Ongoing	23,536	1,636
House 1		24 Sep 2014	02 Jan 2016	16,0275	22,604

Table 9.1: Amount of data obtained in our deployment

Real deployment performed by an external partnership

Before the deployment described above, we had also gathered approximately three months of data from three other houses in France, from a previous deployment in partnership with the Handco company [1]. This dataset had been very instructive and gave us precious feedback for our further deployment.

The amount of data and the list of sensors deployed is summarized in table 9.2.

Residence	**Duration of deployment**	**Number of events**	**Sensors deployed**
House 1	23 May 2012 — 6 Jun 2012	58070	7
House 2	24 May 2012— 18 Jun 2012	86688	6
House 3	4 Jun 2012— 12 Jul 2012	13275	7

Table 9.2: Amount of data obtained in the deployment from Handco

However, this dataset is mentioned in this dissertation solely for informative reasons. In the rest of this section I will rely on the current dataset from Normandie.

9.1.2 Ground-Truth Data Acquired

We have performed a ground-truth acquisition for 6 elderly people living in the nursing home we have deployed, and obtained 465 observations over 10 days, as summarized in Table 9.3. Most of the activities are observed during the morning, as the residents often spend the afternoon in the nursing home's common room. For logistic reasons, we could not observe the ground-truth during evening or night. The human factor also had to be integrated into the process. The first attempt was to ask the nurses to perform punctual acquisitions over days, with only a few observations a day. However, I have faced a low involvement, and did not obtain relevant observations this way. It was then decided to send members of our own research team (myself included) into the nursing home, where we performed the process more intensively, over a short period of time.

A sample of the ground-truth observed for one resident over four days is displayed by Figure 9.2.

Room	**Observations**	**Periods of observation**
Resident 2	122	31 May 2015 — 2 Jun 2015 3 Aug 2015— 4 Aug 2015
Resident 4	36	3 Aug 2015 — 6 Aug 2015
Resident 6	129	31 May 2015 — 2 Jun 2015 3 Aug 2015 — 8 Aug 2015
Resident 7	32	4 Aug 2015 — 8 Aug 2015
Resident 8	53	3 Aug 2015 — 8 Aug 2015
Resident 9	85	3 Aug 2015 — 8 Aug 2015

Table 9.3: Summary of ground-truth observations from our deployments

[1] `http://handco.fr`, I would like to express my personal gratitude to Handco for their involvement and the quality of their work during this partnership.

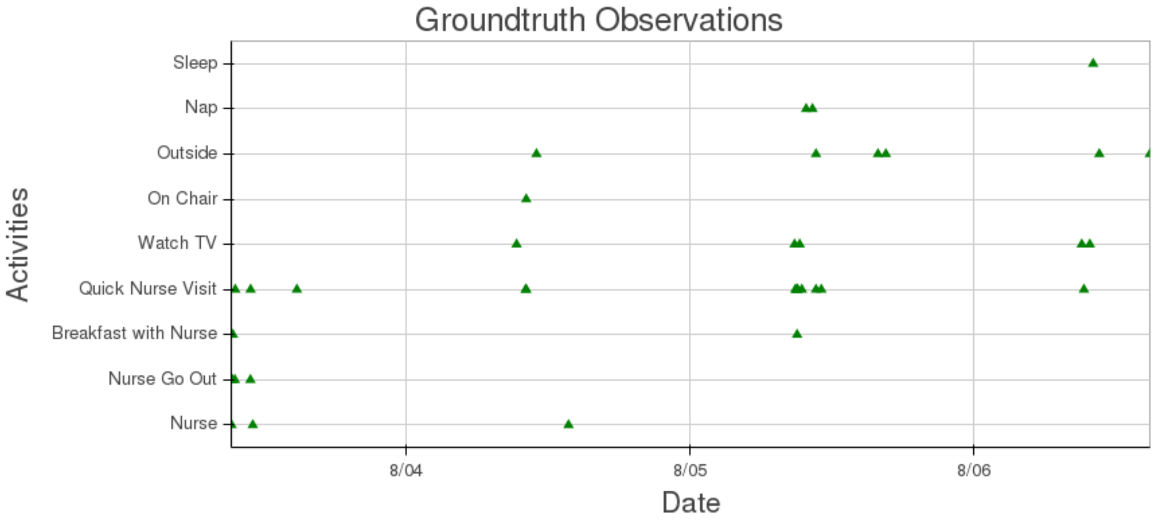


Figure 9.2: Overview of ground-truth for one resident

9.1.3 Feedback from End-Users

During this deployment, we have had frequent contact with the users of the system (i.e., the nursing home staff, and the residents). The nurses have been educated to understand and use the system. The web application has been configured in the nurses' office, and they had access to the UbiSMART services.

Additionally, for the elderly person whose house was deployed, the relatives have also been educated, and received occasional email alerts when a risk was detected, asking them to check the health of the elderly person.

We also had contacts with the elderly people themselves during our visits on-site. These contacts were stimulating, and provided us with a better understanding of the end-users need.

A number of services have been driven by the feedbacks we received from the nursing home staff (e.g., email alerts, visualization of the rooms). We have also adapted our reasoning to the specific conditions of the nursing home: several activities have been discarded (e.g., cooking activities), and others have been added after our observations(e.g., nap).

9.2 Validating Knot as a Prototyping Framework for Activity Recognition

In this section, I propose a functional validation of Knot as a framework for prototyping activity recognition. This validation highlights all the major features of Knot, and validates that consistent results are obtained. The validation is based on the following features:

Validating the simulator: Generating sensor events;

Validating the reasoning engine: Reasoning on generated sensor events;

Validating the data visualization tools: Plotting events and activities.

9.2.1 Validating the Simulator

In order to validate Knot's simulator, I define a scenario, and the simulator will generate a sequence of events matching a pre-defined scenario. The simulator object is binded to the house. The scenario to be executed is described in Table 9.4. In Listing 9.1, I implement the scenario, create the simulator object, and run the simulation.

Activity	**Location**	**Action**	**Duration**
Living Room Activity	Living Room	Moving	1 minute
Bathroom Activity	Bathroom	Moving	40 seconds
Bedroom Activity	Bedroom	Moving	30 seconds
Sleep	Bedroom	Immobile	1 hour
Bedroom Activity	Bedroom	Moving	30 seconds

Table 9.4: Definition of the scenario to be executed

```python
from reasoning.simulator import scenario
from reasoning.simulator.simulator import Simulator

from dateutil.parser import parse

time = parse("2015-09-20 10:00:00")

@scenario("")
def scenario(simulator, env, actor, house):
    actor.set_initial_position("Living Room")
    simulator.set_clock(time)
    yield env.process(actor.go_to("Living Room"))
    yield env.process(actor.keep_moving(timedelta(minutes=1),2))
    yield env.process(actor.go_to("Bathroom"))
    yield env.process(actor.keep_moving(timedelta(seconds=40),6))
    yield env.process(actor.go_to("Bedroom"))
    yield env.process(actor.keep_moving(timedelta(seconds=30),1))
    yield env.process(actor.stay_immobile(timedelta(hours=1)))
    yield env.process(actor.keep_moving(timedelta(seconds=30),3))

simulator = Simulator(house, init_time=time)
simulator.run(scenario, duration=5000)
```

Listing 9.1: Defining a scenario, and running the simulator

From Table 9.5, we observe the events generated by the simulator. These events are fully consistent with the scenario, and we notice that the resident goes by the entrance to move from living room to bathroom, thanks to the smart path planning.

9.2.2 Validating the Reasoning Engine

In order to validate the quality of the reasoning engine, I execute the reasoner on the events generated by the simulator in Table 9.5. The reasoning process is composed of three objects:

– A *source*, to provide the events to the reasoner;

– A *reasoner*, to execute the reasoner from the events it receives;

– A *sink*, to save the inferred activities in a dataframe object for later use.

The three objects are piped together, so that the event source's output are the reasoner's input, and the reasoner's output is the sink's input (Listing 9.2).

resident	date	sensor	value
J	2015-09-20 10:00:02	Living Room Motion	On
	2015-09-20 10:00:12	Living Room Motion	On
	2015-09-20 10:00:22	Living Room Motion	On
	2015-09-20 10:00:32	Living Room Motion	On
	2015-09-20 10:00:42	Living Room Motion	On
	2015-09-20 10:00:52	Living Room Motion	On
	2015-09-20 10:01:02	Living Room Motion	On
	2015-09-20 10:01:12	Living Room Motion	On
	2015-09-20 10:01:18	Entrance Motion	On
	2015-09-20 10:01:34	Bathroom Motion	On
	2015-09-20 10:01:46	Bathroom Motion	On
	2015-09-20 10:01:58	Bathroom Motion	On
	2015-09-20 10:02:10	Bathroom Motion	On
	2015-09-20 10:02:12	Living Room Motion	Off
	2015-09-20 10:02:18	Entrance Motion	Off
	2015-09-20 10:02:22	Bathroom Motion	On
	2015-09-20 10:02:38	Bedroom Motion	On
	2015-09-20 10:02:48	Bedroom Motion	On
	2015-09-20 10:02:58	Bedroom Motion	On
	2015-09-20 10:03:08	Bedroom Motion	On
	2015-09-20 10:03:22	Bathroom Motion	Off
	2015-09-20 10:04:08	Bedroom Motion	Off
	2015-09-20 11:03:17	Bedroom Motion	On
	2015-09-20 11:03:29	Bedroom Motion	On
	2015-09-20 11:03:41	Bedroom Motion	On
	2015-09-20 11:04:41	Bedroom Motion	Off

Table 9.5: Events generated by the simulator

```python
from reasoning.sources.dataframe import event_source
from reasoning.engines.semantic.reasoner import SemanticReasoner
from reasoning.sinks.dataframe_sink import ActivityAsDataframe

engine = SemanticReasoner.build_reasoner(house)
sink = ActivityAsDataframe()

event_source(engine.reason(sink(house)),
             simulator.logger.generated_events.loc['J'])
```

Listing 9.2: Running the reasoner

Table 9.6 shows the inferred activities, given the generated events. These inferred activities are largely matching the ones defined in the scenario (Table 9.4).

resident	date	since	activity	*Correct*
J	2015-09-20 10:01:12	2015-09-20 10:00:02	Living Room Activity	✓
	2015-09-20 10:01:28	2015-09-20 10:01:18	Unknown (Entrance)	∅
	2015-09-20 10:02:32	2015-09-20 10:01:34	Activity in Bathroom	✓
	2015-09-20 10:02:48	2015-09-20 10:02:38	Sleep	✗
	2015-09-20 10:03:42	2015-09-20 10:02:58	Bedroom Activity	✓
	2015-09-20 11:03:39	2015-09-20 10:03:52	Sleep	✓
	2015-09-20 11:04:11	2015-09-20 11:03:41	Bedroom Activity	✓
	2015-09-20 11:04:41	2015-09-20 11:04:21	Sleep	✗

Table 9.6: Activities inferred from the simulated events

9.2.3 Validating the Data Visualisation Tools

In order to validate Knot's data visualization tool, we will visualize activities and events from an existing dataset. To do so, we load events from the CSV file in Listing 9.3, rename the resident "8" into "J", and replaces sensor codes with more explicit names (Listing 9.4). Table 9.7 shows a subsets of the loaded events after the pre-processing.

A similar process is done with the activity dataset, and is not displayed here for brevity purpose.

```
date,                        resident,  sensor,    value
2014-09-23 13:33:00.000000,  1,         A1,        on
2014-09-23 13:33:00.000000,  2,         A1,        on
2014-09-23 13:37:00.000000,  3,         A1,        on
2014-09-23 13:44:40.000000,  5,         C2,        On
2014-09-23 14:05:36.000000,  3,         a819880,   normal
2014-09-23 14:05:42.000000,  5,         C2,        On
2014-09-23 14:05:49.000000,  5,         C2,        On
2014-09-23 14:05:53.000000,  3,         a819880,   alert
2014-09-23 14:05:55.000000,  3,         a819880,   normal
2014-09-23 14:05:57.000000,  3,         a819880,   alert
2014-09-23 14:05:59.000000,  3,         a819880,   normal
2014-09-23 14:06:07.000000,  3,         a819880,   alert
2014-09-23 14:06:10.000000,  3,         a819880,   normal
2014-09-23 14:06:15.000000,  3,         a819880,   alert
2014-09-23 14:06:32.000000,  5,         C1,        On
2014-09-23 14:06:36.000000,  3,         a819880,   normal
2014-09-23 14:06:37.000000,  3,         a819880,   alert
2014-09-23 14:06:38.000000,  3,         a819880,   alert
2014-09-23 14:07:00.000000,  3,         A1,        On
2014-09-23 14:07:06.000000,  3,         A1,        On
```

Listing 9.3: Input events file

```
from reasoning.data_analysis.data_toolbox import load_events
from reasoning.data_analysis.preprocessing import \
    (update_residents,rename_sensors,rename_sensors_from_pattern)

events = load_events('dataset/event_ubismart.csv')

residents = {'J': (8, None, None)}

house_sensors = {'J': {'A1': 'Bedroom Motion',
                       'A2': 'Kitchen Motion',
                       'A3': 'Bathroom Motion',
                       'A4': 'Living Room Motion',
                       'A5': 'Entrance Motion',
                       'A6': 'Living Room Motion',
                       'aFAF300': 'Main Door Sensor',
                       'aD28E00': 'Fridge Door Sensor'}}

events = update_residents(events, residents)
rename_sensors(events, house_sensors)
```

Listing 9.4: Preprocessing events

Listing 9.5 illustrates the `plot` method of Knot, which is used to visualize events and activities. Figure 9.3 displays the output of this method, that is a plot of the events from Table 9.7 and the corresponding activities.

```
from reasoning.data_analysis.data_viz import plot

plot(events=events, activities=sink2.activities, residents=['J'],
    title="", date_min='2015-02-11 11:00:00',
    date_max='2015-02-11 11:25:00')
```

Listing 9.5: Plotting events and inferred activities

9.2.4 Discussion

In this section, I have illustrated the capability of Knot to analyse data and prototype an Activity Recognition reasoning engine. This functional validation validates the soundness of Knot as a prototyping tool. The scientific contributions of this thesis validate the usefulness of Knot, as they would have not been possible without such a tool. The question of whether Knot satisfies the developers and researchers' need cannot yet be generalized beyond this thesis work, as it is still in an early phase of development.

resident	date	sensor	value
J	2014-11-24 17:13:25	Living Room Motion	On
	2014-11-24 17:13:27	Living Room Motion	On
	2014-11-24 17:13:29	Kitchen Motion	On
	2014-11-24 17:13:31	Living Room Motion	On
	2014-11-24 17:13:37	Kitchen Motion	On
	2014-11-24 17:13:44	Kitchen Motion	On
	2014-11-24 17:13:51	Kitchen Motion	On
	2014-11-24 17:14:06	Bedroom Motion	Off
	2014-11-24 17:14:27	Living Room Motion	Off
	2014-11-24 17:14:29	Living Room Motion	Off
	2014-11-24 17:14:55	Kitchen Motion	Off
	2014-11-24 17:15:07	Kitchen Motion	On
	2014-11-24 17:15:36	Kitchen Motion	On
	2014-11-24 17:15:53	Kitchen Motion	On
	2014-11-24 17:16:05	Kitchen Motion	On
	2014-11-24 17:16:30	Bedroom Motion	On
	2014-11-24 17:16:42	Bedroom Motion	On
	2014-11-24 17:16:49	Bedroom Motion	On
	2014-11-24 17:16:56	Bedroom Motion	On
	2014-11-24 17:17:04	Kitchen Motion	On

Table 9.7: Events from dataset

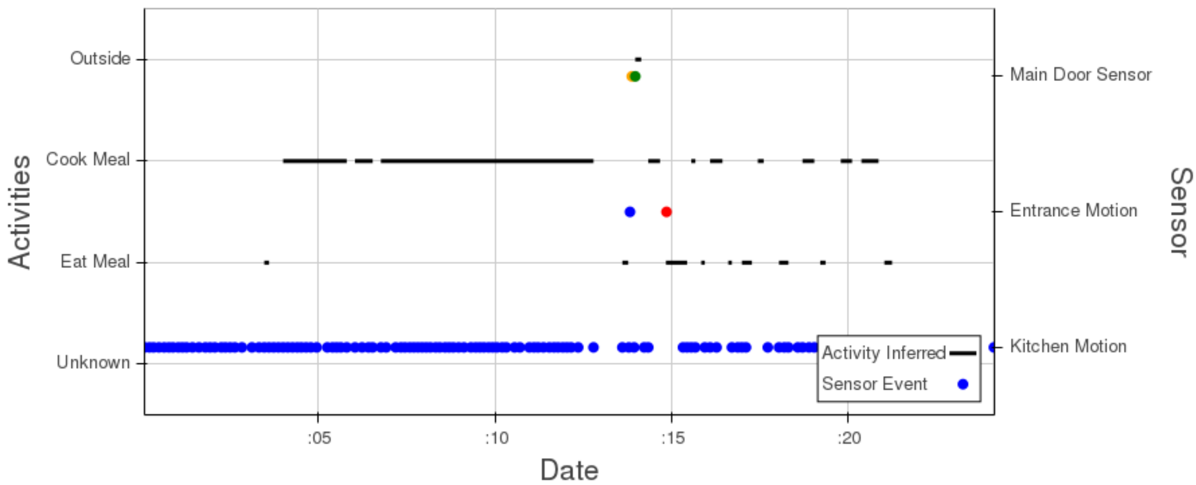


Figure 9.3: Plotting events and inferred activities

At the moment, Knot can be considered as an alpha version, and is still in alpha phase.

9.3 Validating Location Tracking and Multiuser Detection

I have applied the method for location tracking and multiuser detection (Chapter 6) to the individual house we have deployed, where the resident used to receive regular visits. Figure 9.4 shows multiuser situations detected over two days. We observe a large number of noises, spanning multiuser situations over a few minutes. Except for these noises, we detect a multiuser situations on the two morning (certainly caused by a caregiver coming for hygiene), and on the evening before sleep (that would be caused by the visit of a family member). There are long presences in the entrance on the two days, in the early afternoon; these could be caused by the resident being outside, and enforce the importance of integrating door sensors in the algorithm. Apart from these faulty presence in the entrance, the location tracking is clear and realistic, and highlight the life pattern of the resident. Although location tracking does not replace Activity Recognition, it might serve as high-level source of information for a clustering algorithm or an Activity Recognition engine.

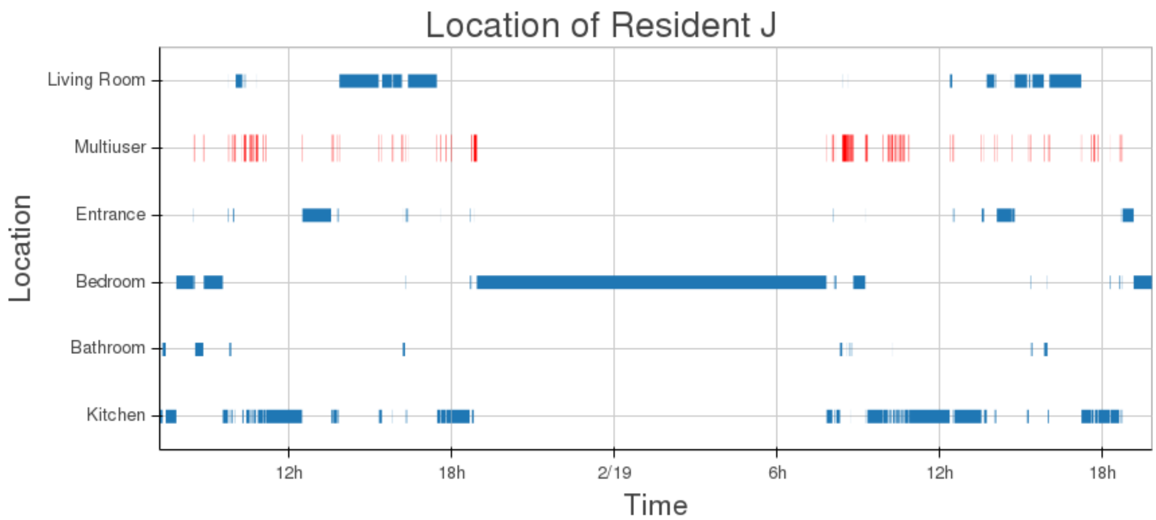


Figure 9.4: Multiuser Situations Over Two Days

Additionally, Figure 9.5 shows the density of multiuser situations by times of the day. We observe that frequent visits occur at 9am (certainly for morning hygiene care), at noon (when the caregiver comes to prepare lunch), and in the evening (from the family). These insights fit the routine this person used to have, based on her family's description. On average, the system estimates that this person spent one hour per day with other people.

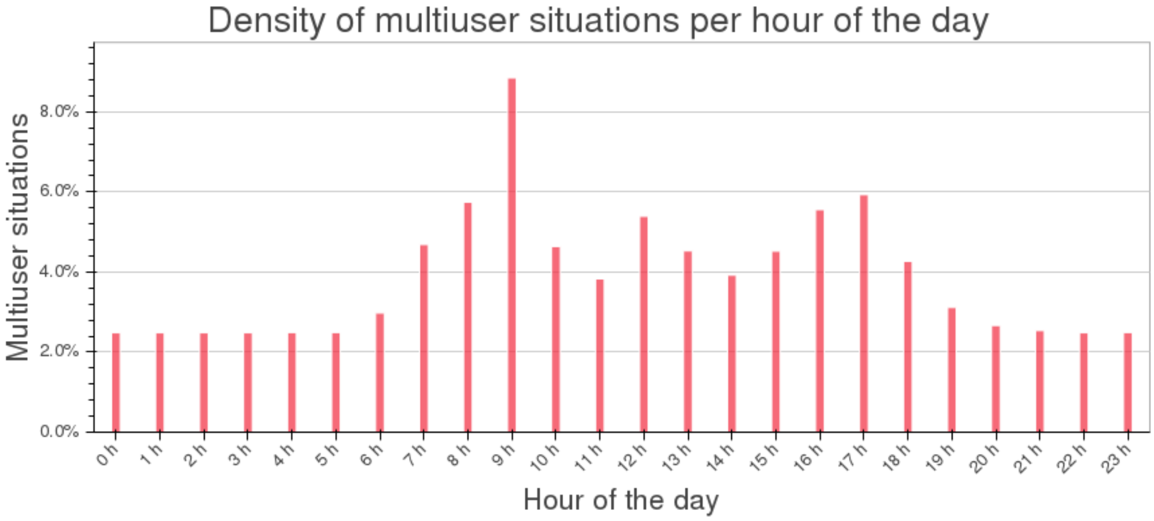


Figure 9.5: Density of Multiuser Situations During the Day

9.4 Validating the Improvements on Decision-Making

In Chapter 7, I have proposed a method to improve the Accuracy of reasoning engines when performing Activity Recognition. Following, I discuss in details the validation of this method.

9.4.1 Updating the Reasoning Engine

This validation is based on Knot's reasoning engine, which is itself based on the UbiSMART reasoning engine. The original reasoning engine does not include the hierarchical activities approach. Moreover, it embedded an arbitrary Score in the reasoning engine. Thus, I have made several improvements to the reasoning engine, that can be summarized as follows:

Introducing the Hierarchical Activities Method
Since UbiSMART is ontology-based, I have formally defined the concepts of *generalization* and *specialization* into the ontology (Section 7.4.2). I have also adapted the rule engine, so that it infers on parent activities first, and iterates recursively to the children activities.

Introducing Activities Scores
The first UbiSMART reasoner applied arbitrary Scores to activities. I change that by calculating Scores of each activity based on the Score measurement method described in 7.2 from Chapter 7, given the Accuracy measured from our ground-truth. Using this Score, the reasoner can conclude on the most reasonable activity (i.e., the activity that has the maximal Score among all the inferred activities).

9.4.2 Acquiring Data

I have experimented the system in a laboratory setup, for validation purpose. I have recreated a house environment (Table 9.8) and defined activity scenarios (Table 9.9). A person playing the role of an end-user has executed each scenario five times, for a total of 45 acquisitions.

Room	**Sensors**
Bedroom	Motion sensor
Bathroom	Motion sensor
Living Room	Motion sensor
Kitchen	Motion sensor
	Door sensor (fridge)
Entrance	Motion sensor
	Door sensor (main door)

Table 9.8: Topology of the experiment

9.4.3 Running The Reasoner

For this validation, I use the same datasets (Section 9.4.2) to run the reasoning engine, before and after applying the proposed method. The reasoner can be executed in two different ways.

1. In production mode, we execute the reasoner in order to infer a single activity at a given time. In this case, Accuracy has to be predefined. This is the mode used for the UbiSMART system to deliver services.

2. In calibration mode, we execute the reasoner in order to measure Accuracy of all activities in hierarchical chain. We expect the reasoner to return all possible valid inferred activities, without making a conclusion. The result of a calibration mode should be used as an input for the production mode. The calibration mode is run only at the first phases of deployment. Once the reasoner has converged towards stationary Accuracy values, there is no need to run calibration mode anymore.

9.4.4 Introducing Validation Metrics

I propose four metrics to validate the proposed method. The first metric is the *total measured Accuracy* (T) vs. ground-truth (Equation 9.1). T is useful to measure the exact Accuracy of the reasoning, given that we have a ground-truth on the executed dataset. T is similar to the measured Accuracy of an activity (Equation 7.1), but it is measured on all activities at once, not on a specific activity. It is defined as the number of times the system inferred correctly (based on ground-truth observations), divided by the total number of activities that are both inferred and observed.

$$T(activities) = \frac{|\{groundtruth(a) = inferred(a)|a \in activities\}|}{|groundtruth(activities) \cap inferred(activities)|} \tag{9.1}$$

Three other metrics are the average value of Accuracy ($\bar{A}$), Precision ($\bar{P}$) and Score ($\bar{S}$) for the inferred activities (Equation 9.2). These three metrics provide an estimation of Accuracy, Precision and Score of reasoning engine in an dataset, even in the absence of ground-truth. $\bar{A}$ is not to be confused with the total measured Accuracy T. $\bar{A}$ is an indicator that can be obtained at any time, whereas T is exact, but requires ground-truth to be measured.

$$\begin{aligned}
\bar{A}(activities) &= \frac{\sum\limits_{a \in activities} (accuracy\,(a) \times duration\,(a))}{\sum\limits_{a \in activities} duration\,(a)} \\
\bar{P}(activities) &= \frac{\sum\limits_{a \in activities} (precision\,(a) \times duration\,(a))}{\sum\limits_{a \in activities} duration\,(a)} \\
\bar{S}(activities) &= \frac{\sum\limits_{a \in activities} (score\,(a) \times duration\,(a))}{\sum\limits_{a \in activities} duration\,(a)}
\end{aligned} \tag{9.2}$$

9.4.5 Results

After running the reasoner in both modes (i.e., calibration and production), Table 9.9 presents Accuracy, Precision and Score values for each activity, after the calibration mode. Score is calculated using Equation 7.2 from Chapter 7, with an Accuracy weight $A = 3$ and a Precision weight $P = 1$. For comparison, a second score is calculated, using $A = 1$ and $P = 1$. We observe that activities that are Imprecise but extremely Accurate, such as "*Kitchen Activity*", are more valued with a higher value of A. With $A = 3$, it has a Score of 79.5%, whereas it only has a Score of 63.2% with $A = 1$. On the opposite, a more Precise and less Accurate activity, such as "*Cook Meal*" has a Score of 64.5% with $A = 3$, and 69.3% with $A = 1$. When both "*Kitchen Activity*" and "*Cook Meal*" are inferred, the reasoner will conclude with "*Kitchen Activity*" if $A = 3$, and with "*Cook Meal*" if $A = 1$.

I run the reasoner four times in production mode: with $(A = 3, P = 1)$ and with $(A = 1, P = 1)$, in both case *before* and *after* running the calibration (Table 9.10). Without calibration, each activity has a default accuracy of 10%, and Scores are calculated accordingly. With $(A = 3, P = 1)$, we measure $T = 93.8\%$ after the calibration, whereas T was only 63.4% before the calibration (+30.4%). $\bar{P}$ has decreased from 6.75 to 4.15

Activity	**Accuracy**	**Precision**	**Score**	
			A=3, P=1	**A=1, P=1**
Kitchen Activity	100%	4	79.5%	63.2%
Eat Meal	0%	8	0%	0%
Cook Meal	60%	8	64.5%	69.3%
Activity in Bathroom	100%	4	79.5%	63.2%
Go Toilet	50%	8	56.2%	63.2%
Hygiene	0%	7	0%	0%
Living Room Activity	100%	4	79.5%	63.2%
Read Book	50%	8	56.2%	63.2%
Watch TV	0%	8	0%	0%
Bedroom Activity	100%	6	88%	77.5%
Sleep	100%	7	91.5%	83.7%
Outside	100%	7	91.5%	83.7%
Come Home	10%	7	16.3%	26.5%

Table 9.9: Scores of the reasoner

(-2.60), and $\bar{A}$ has increased from 55.5% to 89.4% (+33.9%). This illustrates the trade-off between Accuracy and Precision mentioned in Section 7.1. With ($A = 1, P = 1$), the calibration impact is less significant. T happens to increase by 3.3%. $\bar{A}$ decreases by 11.8%, and $\bar{P}$ increases by 0.26. This is explained by the fact that with $A = 1$, the reasoner is not allowed to decrease much Precision in favor of Accuracy. Thus, it will tend to be as Precise as possible, which is similar to its default behavior, without the method introduced in this paper. Finally, one may notice that $\bar{S}$ increases with both values of (A, P): +16.5% with ($A = 3, P = 1$) and +1.7% with ($A = 1, P = 1$). This is inherent to the method I propose, which always selects the activity with the maximal Score among all the inferred activity.

Sample	**A**	**P**	**Total Measured Accuracy**	**Average Accuracy**	**Average Precision**	**Average Score**
Before calibration	*3*	*1*	63.4%	55.5%	6.75	56.9%
After calibration			93.8%	89.4%	4.15	73.4%
Before calibration	*1*	*1*	63.4%	67.3%	5.56	59%
After calibration			66.7%	55.5%	5.82	60.7%

Table 9.10: Results with ($A = 3, P = 1$) and ($A = 1, P = 1$), *before* and after the calibration

9.5 Validating the Detection of Problematic Reasoning Rules

In this section, I propose a functional validation of the proofs-of-concept described in Chapter 8, by executing a minimal working example for each of features, namely the *houses factory*, the *rules factory*, the detection of missing sensors and the detection of conflicting rules.

9.5.1 Validating the Houses Factory

For this validation, let us assume we need to create the house described in Table 9.11. Listing 9.6 shows the creation of a house object, add the rooms, the doors, a fridge item, and all the sensors, using JSON format.

Room	**Sensor**	**Doors to**
Living Room	Motion Sensor	Entrance
Bedroom	Motion Sensor	Entrance
Bathroom	Motion Sensor	Entrance
Kitchen	Motion Sensor Contact Sensor (fridge)	Living Room
Entrance	Motion Sensor Contact Sensor (main door)	Outside

Table 9.11: Definition of the house

```
[{ "house_name": "J",
    "rooms": [
        {"name": "Bedroom", "type": "bedroom"},
        {"name": "Bathroom", "type": "bathroom"},
        {"name": "Kitchen", "type": "kitchen"},
        {"name": "Entrance", "type": "entrance"},
        {"name": "Living Room", "type": "livingRoom"}],
    "doors": [
        {"doorFrom": "room:Living Room", "doorTo": "room:Entrance"},
        {"doorFrom": "room:Bedroom", "doorTo": "room:Entrance"},
        {"doorFrom": "room:Bathroom", "doorTo": "room:Entrance"}
        {"doorFrom": "room:Kitchen", "doorTo": "room:Living Room"}
        {"doorFrom": "room:Entrance", "doorTo": "Outside"}],
    "items": [
        {"type": "fridge", "name": "Fridge", "locatedIn": "room:Kitchen"}],
    "sensors": [
        {"code": "A1", "type": "Motion Sensor",
            "attachedTo": "room:Bedroom"},
        {"code": "A2", "type": "Motion Sensor",
            "attachedTo": "room:Bathroom"},
        {"code": "A3", "type": "Motion Sensor",
            "attachedTo": "room:Kitchen"},
        {"code": "A4", "type": "Motion Sensor",
            "attachedTo": "room:Entrance"},
        {"code": "A5", "type": "Motion Sensor",
            "attachedTo": "room:Living Room"},
        {"code": "C1", "type": "Contact Sensor",
            "attachedTo": "item:Fridge"},
        {"code": "C2", "type": "Contact Sensor",
            "attachedTo": "door:Entrance:Outside"}],
    "persons": [{"name": "John Doe", "type": "Resident"}]
}]
```

Listing 9.6: Defining the house, using House Factory

The house definition in Listing 9.6 generates the ontology in Listing 9.7. This ontology is absolutely equivalent to what it would have been if it had been written manually. Defining a house through the house factory is more intuitive, more concise and less error-prone than writing the ontology manually. In this case, the house is defined in 28 lines of code when using the house factory, for a generated ontology of 73 lines of code.

```
# ...
hom:door_corridor-bathroom a qol:Door ;
    qol:doorFrom hom:room_corridor ;
    qol:doorTo hom:room_bathroom .
# ...

hom:resident_johnDoe a qol:Resident ;
    qol:detectedIn hom:outside ;
    qol:name "John Doe" ;
    qol:residentIn hom:house_j .
# ...

hom:sensor_bedroomMotion a qol:MotionSensor ;
    qol:deployedIn hom:room_bedroom ;
    qol:id "Bedroom Motion"@en .
# ...

hom:item_fridge a qol:Fridge ;
    rdfs:label "Fridge"@en ;
    qol:locatedIn hom:room_kitchen .
# ...

hom:house_j a qol:House ;
    qol:name "J" .
# ...
```

Listing 9.7: Generated ontology of the house

9.5.2 Validating the Rules Factory

In order to demonstrate the rules factory, I create a basic set of rules for recognizing activities during the reasoning process (Table 9.12).

These rules are implemented in Listing 9.8, using the `house.add_activity(name, conditions)` method.

Activity	Conditions
Living Room Activity	Detected in Living Room
Activity in Bathroom	Detected in Bathroom
Kitchen Activity	Detected in Kitchen Motion ≥ 2
Bedroom Activity	Detected in Bedroom Motion ≥ 3
Sleep	Detected in Bedroom Duration $\geq$ 10 seconds Duration $\leq$ 8 hours Motion ≤ 2
Outside	Detected in Outside
Come Home	Detected in Entrance Comes From Outside

Table 9.12: Definition of the activities

```
[
    {"name": "Living Room Activity",
        "conditions": {"detected_in": {"room": "Room:Living Room"}}},
    {"name": "Kitchen Activity",
        "conditions": {"detected_in": {"room": "room:Kitchen",
                                       "min_motion": 2}}},
    {"name": "Activity in Bathroom",
        "conditions": {"detected_in": {"room": "room:Bathroom"}}},
    {"name": "Sleep",
        "conditions": {"max_duration": "8h",
                       "detected_in": {
                            "room": "room:Bedroom",
                            "min_duration": "10s",
                            "max_motion": 2}}},
    {"name": "Outside",
        "conditions": {"detected_in": {"room": "room:Outside"}}},
    {"name": "Come Home",
        "conditions": {"detected_in": {"room": "room:Entrance"},
                       "coming_from": "room:Outside"}},
    {"name": "Bedroom Activity",
        {"detected_in": {"room": "room:Bedroom",
                         "min_motion": 3}}}
]
```

Listing 9.8: Defining activities, using Rule Factory

Listing 9.9 shows the generated set of N3 rules. Again, the rules factory offers a more intuitive, concise, and less error-prone interface to define rules, rather than writing them manually in N3 format. In this case, the rules are defined in 21 lines of code when using the rule factory, and the generated ontology takes 84 lines of code.

```
{
    ?u qol:detectedIn ?r ;
        qol:inRoomFor ?dr ;
        qol:residentIn hom:house_j .
    hom:clock qol:hasValue ?now .
    ?dr math:notGreaterThan 3.155379e+11 .
    ?r a qol:Bathroom ;
        qol:motionMeasured ?mm .
} => { ?u qol:couldBeDoing hom:activityInBathroom . } .
{
    ?u qol:detectedIn ?r ;
        qol:inRoomFor ?dr ;
        qol:residentIn hom:house_j .
    hom:clock qol:hasValue ?now .
    ?dr math:notGreaterThan 3.155379e+11 .
    ?mm math:notLessThan 3 .
    ?r a qol:Bedroom ;
        qol:motionMeasured ?mm .
} => { ?u qol:couldBeDoing hom:bedroomActivity . } .
# ...
```

Listing 9.9: Generated ontology of the activities

9.5.3 Validating the Detection of Missing Sensors

Following, I validate the detection of missing sensor, using a minimal environment defined in Tables 9.13 and 9.14. In this environment, the “Sleep” activity cannot be recognized without a motion sensor in the bedroom, and the “Nap” activity cannot be recognized without the “Sleep” activity being recognized first. The “Prepare Lunch” activity depends on the fridge, thus cannot be recognized without a sensor on the fridge. On the other hand, the “Go Toilet” activity can be inferred correctly, because we declare a motion sensor in toilet. These inconsistencies have been correctly inferred by the proposed algorithm (Table 9.15).

Room	Items	Sensors
Bedroom	$\emptyset$	$\emptyset$
Kitchen	Fridge	A1
Toilet	$\emptyset$	A2

Table 9.13: Detecting missing sensors — Defining a house

Activity	Conditions	Value	Parent Activity
Sleep	Detected In	Bedroom	∅
	Max Motion	1	
Nap	Hour of the day	13h — 18h	Sleep
Go Toilet	Detected In	Toilet	∅
Prepare Lunch	Detected In	Kitchen	∅
	Using Item	Fridge	

Table 9.14: Detecting missing sensors — Defining rules

Activity	Expected Issues	Detected Issues	Correct
Sleep	Detected In(Bedroom)	Detected In(Bedroom)	✓
Nap	Parent(Sleep)	Parent(Sleep)	✓
Go Toilet	∅	∅	✓
Prepare Lunch	Using Item(Fridge)	Using Item(Fridge)	✓

Table 9.15: Detecting missing sensors — Results

9.5.4 Validating the Detection of Conflicting Sensors

The following example demonstrates the validity of the proposed method for detecting of conflicting rules (Table 9.16). If the resident is in the kitchen at 13:00, he might either be eating lunch or washing the plates. However, there is no conflict between nap and night sleep, as the two activities occur at distinct times.

9.6 Discussion

In this chapter, I have validated the five major contributions of my thesis work. These are:

– UbiSMART, a production-ready framework for AAL;

– Knot, a prototyping tool for Activity Recognition and data analysis in the context of AAL;

– An algorithm for tracking location of the resident in a smart home, and detect multiuser situations;

– A method for improving the Accuracy of Activity Recognition reasoning engines, by introducing a more rational decision-making process.

– Two methods to detect inconsistencies in rule-based reasoning engines.

For each of these contributions, I have made efforts to perform a thorough validation, taking into account the constraints inherent to the contribution (i.e., a framework's validation tends to be qualitative, whereas improvements related to Activity Recognition are quantitative). Generally speaking, the results found in this validation are satisfying:

<table>
<tr><th>Rule</th><th>Conditions</th><th>Detected Overlapping situations</th><th>Correctly detected</th></tr>
<tr><td>Read Book</td><td>Detected in Living Room

Motion ≤ 2</td><td rowspan="2">Detected in Living Room

Motion ≤ 2</td><td rowspan="2">✓</td></tr>
<tr><td>Watch TV</td><td>Detected in Living Room</td></tr>
<tr><td>Eat lunch</td><td>Detected In Kitchen
11:00 $\leq$ time $\leq$ 14:00</td><td rowspan="2">Detected in Kitchen

13:00 $\leq$ time $\leq$ 14:00</td><td rowspan="2">✓</td></tr>
<tr><td>Wash the plate</td><td>Detected In Kitchen
13:00 $\leq$ time $\leq$ 15:00</td></tr>
<tr><td>Night Sleep</td><td>Detected in Bedroom
Motion ≤ 1
20:00 $\leq$ time $\leq$ 10:00</td><td rowspan="2">$\emptyset$</td><td rowspan="2">✓</td></tr>
<tr><td>Nap</td><td>Detected in Bedroom
Motion ≤ 1
13:00 $\leq$ time $\leq$ 18:00</td></tr>
</table>

Table 9.16: Examples of conflicting rules detection

- The UbiSMART framework is stable, deployed, and provides data continuously;
- Knot performs accordingly to its specifications;
- Unfortunately, I have not obtained significant ground-truth to perform a quantitative validation of the multiuser detection algorithm, although the output is encouraging, from a qualitative point-of-view.
- I have obtained significant improvements in the Accuracy of Activity Recognition reasoning engines.

Part VI

Conclusion

It's a dangerous business going out of your door. You step into the road, and if you don't keep your feet, there is no knowing where you might be swept off to.

Bilbo Baggins

10 Conclusion

This thesis studies the requirements that Ambient Assisted Living (AAL) systems should fulfill in order to extend into large-scale deployments. It focuses on the problematics of making Activity Recognition more reliable and assisting the expert's work when defining activities. While AAL researches slowly shift towards real deployments, I hope to encourage this shift by showing the benefits such real deployment had on my thesis.

10.1 General Approach Selected In This Thesis

The key to this thesis research was to "get to the real world", that is to deploy our system in a real environment. More than just validating my contributions, this deployment has guided my research, and helped me to adopt a user-centric approach.

During the development of this thesis, I have deliberately chosen to broaden my contributions to two distinct layers: the **improvement of Activity Recognition accuracy** and the development of **prototyping tools for Activity Recognition**. The core activity of my thesis was indeed to answer to its main problematic (i.e., making Activity Recognition more reliable). By doing so, I have observed that only little infrastructure was available to prototype on Activity Recognition. Thus, I took the decision to extend my thesis scope, by developing powerful prototyping tools for supporting my research on Activity Recognition, as well as other researchers' further researches.

For AAL prototyping, I encourage the development of semi-autonomous tools, which perform calculations and report relevant informations to the expert, similar to services offered by Business Intelligence. I believe the key to an effective prototyping workflow are the following:

- Support short and reactive prototyping cycles, featuring rapid feedbacks;
- Merge powerful models provided by ontologies with data-analysis tools:
    - Provide data analysis and visualization tools for obtaining feedback;

  - Lower the technical cost of working with ontological model;

- Automate the installation steps, to make the environment reproducible between developers;

- Provide an interactive development environment (e.g., development notebooks rather than traditional packages).

As for Activity Recognition, I encourage a process of continuous improvement, using the prototyping workflow described above. This paradigm of continuous improvement must be supported by a metric to measure the quality of an Activity Recognition reasoning engine. Therefore I have introduced in this thesis the concept of Score, based on accuracy and precision of activities.

Generally speaking, I believe that AAL should be guided by rationality, which I define as "taking what is probably the best decision, given the available knowledge of the context".

10.2 Contributions

Scientific contributions

My scientific contributions can be summarized as follows:

- I have created an algorithm to detect multi-users situations (Chapter 6). These situations are rather easy to detect, but complicated to handle, as they confuse the sensors by creating interwoven activities. Fortunately, we have stated that if the resident is having a visitor, he does not require our system to be active. We can therefore stop the reasoning for the time being, until the resident is alone again.

- I have introduced a method for measuring the accuracy of a reasoning engine when inferring an activity (Section 7.4.2 from Chapter 7). The final result is a Score attached to a rule from our reasoning engine.

- I have introduced a hierarchical model for activities (Section 7.4.2 from Chapter 7). This model is better suited to define activities at different levels of Precision, without conflict or overlapping.

- I have then introduced a method for optimizing the decision making process in Activity Recognition, based on precision and accuracy (Chapter 7). This is the major contribution from my thesis. It benefits directly from the hierarchical model described above. It consists into tagging each activity with a precision value (defined by the expert) and a confidence value (from our accuracy measurement). The reasoning engine then choses among every possible activity, the one that has the best precision-accuracy couple.

– I have developed two algorithms to detect inconsistencies in the definition of rules for activity recognition. The first algorithm detects rules that will never trigger because they depend on a sensor that is not part of the environment. Thus, if an activity requires a certain set of sensors to be recognized, but one or more of these sensors are absents, the system can detect that the activity can never be inferred. The system can then alert the technician, and suggest him to to install new sensors. The second algorithm detects possible conflicts between rules (i.e., situations where multiple activities could be recognized, with no way to select one over the others).

Technical Contributions

In addition to the scientific contributions listed above, several technical requirements emerged from our deployments, and have been addressed. These technical contributions are extensively detailed in Chapter 4, and can be summarized in two parts. The first part is the implementation of the Ubiquitous Service MAnagement & Reasoning archiTecture (UbiSMART) framework, as a web application for visualizing activities, running the reasoning process, delivering services and handling the different users of our system. The second part (to which I contributed most) is the gateway, through the Ubigate application. Gateways are the core of a smart-home, they play the role of bridge between sensors and devices in the house, and the central server. Due to numerous constraints from within the smart home (such as the absence of Internet connection), a particular focus has been given to reliability and maintenance. Additionally, I have worked on easy reproducibility of our system for large-scale deployments in a wide number of houses.

Another technical contribution, at the source of every scientific contributions listed above, is the creation of Knot (Chapter 5): a proper development environment for data analysis and prototyping; it includes facilities to load the datasets, a framework for analysing data, a dynamic definition of smart-home environments through the houses factory and the rules factory, and visualization tools.

10.3 Results

As an outcome of this thesis, my team and I have deployed our system (UbiSMART) in a real-world environment. The relation of this deployment with my thesis work is bidirectional. Indeed, my research has been supporting the deployment, while the deployment has been guiding my research. At the present moment, the deployment is still ongoing, and used by the end-users.

I have also validated the usage of Knot (the prototyping framework I have developed) in a straightforward way: I have used Knot myself for developing my own contributions to Activity Recognition.

Most importantly, I have effectively improved the accuracy of our Activity Recognition reasoning engine with the method I have proposed.

10.3.1 Limitations

Looking at the outcome of this thesis, I find three remaining limitations, which it would be necessary to tackle, beyond the scope of this thesis:

- Currently, AAL is at the crossing path between research and industry. There exists no complete AAL system that would be robust enough to be carried forward to industry. Real-world deployments must be restrained to research projects, where the end-users are advised about the relative instability of such projects. Under this situation, large-scale deployments of AAL systems will not yet become a reality.
- Knowledge-driven methods for Activity Recognition will eventually show their limits, when the complexity of AAL systems will become too high for a human to design it. At that moment, researchers will have to integrate data and machine-learning to the design and reasoning process.
- Activity Recognition is only one element in the AAL workflow. It must eventually lead to service delivery features, in order to have a real impact on the end-users.

10.4 Perspectives

The major perspective out of this thesis is to extend our system towards large-scale deployments. This would require our system to become closer to industry, with more robust gateways in the smart-homes. I envision Ubigate — our gateway — as a distribution, where the system could be installed by anyone, and maintain itself through automatic updates. It would include a number of user-contributed packages to deliver various services, depending on use cases. Service provisioning has not been integrated in the scope of this thesis, but remains a key requirement for AAL systems.

The second perspective also derives directly from this thesis work. It would consist into consistently improving the reasoning engine, until obtaining a highly reliable reasoner for activity recognition, with 99% of accuracy even at acceptable precisions. Then, and only then, would AAL be ready to shift completely from research to its industrial applications.

A first step to improve the accuracy of Activity Recognition will be to work further on user-specific reasoning. This means that each end-user should have its own set of rules defined, adapted to his environment and his daily routine. To keep this feature realistic in a large-scale deployment, the reasoning should be able to adapt automatically to a new user, or possibly use crowd-sourcing as a knowledge source.

Another perspective — which has not been integrated to the scope of this thesis — would be to explore the relation between ground-truth data and sensor events. Indeed, ground-truth data could be used as labels for data-driven methods, creating opportunities to use supervised machine learning method.

A final perspective for improving activity recognition would be to create a bridge between knowledge-driven and data-driven approach. This feature would represent a huge step forward for AAL, as it would keep the expressiveness and intuitiveness of knowledge-driven methods, while benefiting from the power of data-driven methods. I argue that the scope of this feature could extend beyond AAL, and be generalized as knowledge engineering. I believe that knowledge engineering will be the keystone for complex model analysis, and will create a revolution that will be known tomorrow as knowledge science, in complement to data science.

Part VII

Appendix

A

Questionnaires — The quality of life of elderly people

This appendix supports the analysis on the needs of elderly people (Chapter 3). It includes the English translation of the questionnaires sent to the elders and to their caregiver.

The habits of life of elderly people

Questionnaire to be completed by the elder

<Translated from French for the purpose of this appendix>

2012 – QoL Research Chair / Institut Mines-Télécom / La Mutuelle Générale / Association Nationale des retraités de la Poste et de France Télécom

The information gathered in this survey will be statistically analysed to better understand the life habits of elderly people. Individual answers will remain anonymous. There are no good or bad answers. The gathered data will be sent to the Handicom laboratory.

According to the law "Informatique et Libertés" from January 6th 1978, modified in 2004, you have a right to access and modify your information, which you may apply by contacting the Handicom laboratory. You can also refuse the treatment of your data.

Handicom laboratory,
Telecom Sud Paris,
9 rue Fourier 91011 Evry

Your situation

1. How old are you?
2. You are: ☐ female ☐ male
3. What are the jobs you have practices in your professional life?
4. What is the highest diploma you have had?
5. Until what age did you go to school?
6. You live:
   ☐ Alone ☐ In Couple ☐ Other (please describe:)
7. You are:
   ☐ Married ☐ Widow ☐ Single
8. You live in:
   ☐ Your own residence
   ☐ A relative's house
   ☐ A specialized institution (e.g., nursing home). Please describe:
9. How many persons live in your residence? ☐
   - Persons aged less than 60 years old ☐
   - Persons aged more than 60 years old ☐
10. You are: ☐ Owner ☐ Tenant
11. You live: ☐ In city ☐ In countryside

12. Do you use any technical aids (glasses, hearing aids, wheelchair, telealarm…)

Technical aids	**Where and when do you use it?**	**Frequency of use (e.g., once a day)**
Glasses		
Hearing aid		
Wheelchair		
Telealarm		
Cane		
Walker		
Other (please describe):		

Your daily activities (at home)

1. What are your main activities? (e.g., watch TV, read, invite friends…)
2. Are there some activities that you did before, and won't do anymore?
   ☐ Yes ☐ No
3. If yes:

Which ones	Why don't you do it anymore?

4. **Eyes.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never do it alone**
Read the newspaper (even with glasses)					
Distinguish the face of a person other side of a room					
Recognize somebody on a picture					

5. **Dressing.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never do it alone**
Put on clothes					
Button his shirt					
Choose clothes based on the time of the day (pyjama by night, coat to go out...)					
Put clothes in the right order (underwear before pants...)					

6. **Preparing meals.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never do it alone**
Follow a recipe					
Decide of a balanced meal					
Know the content of the fridge					
Prepare a meal					
Heat a meal					

7. **House cleaning.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	Often	Sometimes	Rarely	Never	Never do it alone
Use a broom					
Clean dust					
Reorder a room					
Move from one room to another					
Vacuum the floor					

8. **Taking medications.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	Often	Sometimes	Rarely	Never	Never do it alone
Know where he had put his medicines or his prescription					
Know what medication to take, depening on the time of the day					
Know how many pills to take					
Know when he must visit a doctor					
Know if he has enough medicines for the week					

9. **Administrative activities.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	Often	Sometimes	Rarely	Never	Never do it alone
Fill-in a form					
Manage alone the financial matters (pay the bills...)					
Remember his appointments					
Remember what he has to do					

10. **Use of electronic devices.** *Tick and complete the corresponding cells.*
A) Which electronic device do you **own**?

Device owned	Yes	No	If no, why
Oven			
Micro-wave oven			
Fridge			
Vacuum cleaner			
Mixer			
Kettle			
Coffee machine			
Toaster			
Stove			
Television			
Home Phone			
Mobile phone			
Computer			
Internet			
DVD player			
Videotape player			
Music player			
Radio			
Other (please describe):			

B) Which of these devices do you **use**?

Device used	Yes	No	If yes		If no
			For what usage?	Frequency (e.g., once a day)	Why
Oven					
Micro-wave oven					
Fridge					
Vacuum cleaner					
Mixer					
Kettle					
Coffee machine					
Toaster					
Stove					
Television					
Home Phone					
Mobile phone					
Computer					
Internet					
DVD player					
Videotape player					
Music player					
Radio					
Other (please describe):					

C) **Use of mobile phone.** *Tick the corresponding cell for each row.*

Do you have more difficulties than before to:	Often	Sometimes	Rarely	Never	Never do it alone
Hear a discussion on phone					
Follow a discussion on phone					
Dial a new phone number					
Dial a well-known phone number					
Know who he is talking to on phone					

11. **Safety.** Do you feel safe in your house?
☐ Yes ☐ No

If no, why?

Outside activities

1. What are your main activities outside?

2. How often do you perform the following activities:
Fill-in the cells for each row.

	Frequency (e.g., once a day)
Walk	
Shopping	
Grocery store	
Administrative activities	
Visiting a relative (friend, family...)	
Leisure (hairdresser, cinema…)	
Medical appointment	
Others (please describe)	

3. What do you do to prepare your outside activities? (e.g., check the bus schedule, book a taxi, prepare a shopping list…)

4. When you go out, what are the objects you bring? (e.g., mobile phone, cane, list of things to do…)

5. Your vehicles.

Tick the corresponding cell for each row.

How often do you use:	**Often**	**Sometimes**	**Rarely**	**Never**	**Do not know**
Subway					
Bus					
Train					
Tramway					
His own vehicle (he drives)					
The vehicle of a relative (he is passenger)					
Walk					
Motorcycles					
City bus					
Ambulance					
Other (please describe):					

6. Do you face some difficulties while going out?

☐ Yes ☐ No

If yes, which ones?

7. Do you feel safe when you go out?

☐ Yes ☐ No

If no, why?

Your social and financial aids

1. Do you often see your friends and family?

☐ Yes ☐ No

2. How often do you sees your friends and family?

3. Do you think you are surrounded enough by your relatives?

☐ Yes ☐ No

If no, why?

4. Do you use any home services? Describe which ones and the frequency of the services. *Fill-in the corresponding cells for each row.*

	Yes	**No**	**Frequency (e.g. once a day)**
Cleaning service			
Medical service			
Nurse			
Grocery delivery			
Hairdresser			
Beautician			
Pedicure			
Physiotherapist			
Meal delivery			
House maintenance			
Laundry service			
Others (please describe)			

5. If you have no home services, please explain why.

6. What do you think are the advantages of home services?

7. What do you think are the drawbacks of home services?

8. What do you think are important criterias to select a home service?

9. Who pays?

10. Do you receive financial aids? If yes, which one(s)?

11. How many hours are included in your financial aids?

12. Do you have a health insurance? If yes, which one?

In the future

1. Would you like to be contacted again to participate to a second study and test for free an assistance system developed by researchers from the handicom laboratory?

☐ Yes ☐ No

2. If yes, what kind of services seem useful to you in your daily life?

The habits of life of elderly people

Questionnaire to be completed by a caregiver or a family member

<Translated from French for the purpose of this appendix>

2012 – QoL Research Chair / Institut Mines-Télécom / La Mutuelle Générale / Association Nationale des retraités de la Poste et de France Télécom

The information gathered in this survey will be statistically analysed to better understand the life habits of elderly people. Individual answers will remain anonymous. There are no good or bad answers. The gathered data will be sent to the Handicom laboratory.

According to the law "Informatique et Libertés" from January 6th 1978, modified in 2004, you have a right to access and modify your information, which you may apply by contacting the Handicom laboratory. You can also refuse the treatment of your data.

Handicom laboratory,
Telecom Sud Paris,
9 rue Fourier 91011 Evry

Your situation

1. How old are you?
2. You are: ☐ female ☐ male
3. What is your job?
4. How are you related with the person you help?
5. What kind of helps do you bring to him?
6. In average, how many hours per week do you spend helping him?
7. What is the distance between your home and the home of the person you help?
8. Do you think his residence is adapted to his needs?
   ☐ Yes ☐ No

   If not, why?

His daily activities (at home)

1. What are his main activities? (e.g., watch TV, read, invite friends…)
2. Are there some activities that he did before, and won't do anymore?
   ☐ Yes ☐ No
3. If yes:

Which ones	Why doesn't he do it anymore?

4. **Eyes.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Read the newspaper (even with glasses)					
Distinguish the face of a person other side of a room					
Recognize somebody on a picture					

5. **Dressing.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Put on clothes					
Button his shirt					
Choose clothes based on the time of the day (pyjama by night, coat to go out...)					
Put clothes in the right order (underwear before pants...)					

6. **Preparing meals.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Follow a recipe					
Decide of a balanced meal					
Know the content of the fridge					
Prepare a meal					
Heat a meal					

7. **House cleaning.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Use a broom					
Clean dust					
Reorder a room					
Move from one room to another					
Vacuum the floor					

8. **Taking medications.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Know where he had put his medicines or his prescription					
Know what medication to take, depening on the time of the day					
Know how many pills to take					
Know when he must visit a doctor					
Know if he has enough medicines for the week					

9. **Administrative activities.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Fill-in a form					
Manage alone the financial matters (pay the bills...)					
Remember his appointments					
Remember what he has to do					

10. **Telealarm system.** Does he has a telealarm system?

☐ Yes ☐ No

If yes, which ones?

☐ Geo-location
☐ Call center
☐ Fall detection
☐ Other, please explain:

11. **Use of electronic devices.** *Tick and complete the corresponding cells.*

A) Which electronic device does he **own**?

Device owned	**Yes**	**No**	**If no, why**
Oven			
Micro-wave oven			
Fridge			
Vacuum cleaner			
Mixer			
Kettle			
Coffee machine			
Toaster			
Stove			
Television			
Home Phone			
Mobile phone			
Computer			
Internet			
DVD player			
Videotape player			
Music player			
Radio			
Other (please describe):			

B) Which of these devices does he **use**?

Device used	**Yes**	**No**	**If yes**		**If no**
			For what usage?	**Frequency (e.g., once a day)**	**Why**
Oven					
Micro-wave oven					
Fridge					
Vacuum cleaner					
Mixer					
Kettle					
Coffee machine					
Toaster					
Stove					
Television					
Home Phone					
Mobile phone					
Computer					
Internet					
DVD player					
Videotape player					
Music player					
Radio					
Other (please describe):					

C) **Use of mobile phone.** *Tick the corresponding cell for each row.*

Does he have more difficulties than before to:	**Often**	**Sometimes**	**Rarely**	**Never**	**Never does it alone**
Hear a discussion on phone					
Follow a discussion on phone					
Dial a new phone number					
Dial a well-known phone number					
Know who he is talking to on phone					

12. **Safety.** Do you feel that he is safe in his house?

☐ Yes ☐ No

If no, why?

Outside activities

1. What are his main activities outside?

2. How often does he perform the following activities:
Fill-in the cells for each row.

	Frequency (e.g., once a day)
Walk	
Shopping	
Grocery store	
Administrative activities	
Visiting a relative (friend, family...)	
Leisure (hairdresser, cinema…)	
Medical appointment	
Others (please describe)	

3. What does he do to prepare his outside activities? (e.g., check the bus schedule, book a taxi, prepare a shopping list…)

4. When he goes out, what are the objects he brings? (e.g., mobile phone, cane, list of things to do…)

5. His vehicles.
Tick the corresponding cell for each row.

How often does he use:	**Often**	**Sometimes**	**Rarely**	**Never**	**Does not know**
Subway					
Bus					
Train					
Tramway					
His own vehicle (he drives)					
The vehicle of a relative (he is passenger)					
Walk					
Motorcycles					
City bus					
Ambulance					
Other (please describe):					

6. Does he face some difficulties while going out?

☐ Yes ☐ No

If yes, which ones?

7. Do you feel that he is safe when he goes out?

☐ Yes ☐ No

If no, why?

His social and financial aids

1. Does he often see his friends and family?

   ☐ Yes ☐ No

2. How often does he sees his friends and family?

3. Do you think he is surrounded enough by his relatives?

   ☐ Yes ☐ No

   If no, why?

4. Does he use some home services? Describe which ones and the frequency of the services. *Fill-in the corresponding cells for each row.*

	Yes	**No**	**Frequency (e.g. once a day)**
Cleaning service			
Medical service			
Nurse			
Grocery delivery			
Hairdresser			
Beautician			
Pedicure			
Physiotherapist			
Meal delivery			
House maintenance			
Laundry service			
Others (please describe)			

5. If he has no home services, please explain why.

6. What do you think are the advantages of home services?

7. What do you think are the drawbacks of home services?

8. What do you think are important criterias to select a home service?

9. Who pays?

10. Does he receive financial aids? If yes, which one(s)?

11. How many hours are included in his financial aids?

12. Does he have a health insurance? If yes, which one?

In the future

1. Would you like to be contacted again to participate to a second study and test for free an assistance system developed by researchers from the handicom laboratory?

   ☐ Yes ☐ No

2. If yes, what kind of services seem useful to you in your daily life?

B Location Detection in Real Condition — Implementation

This Appendix supports Chapter 6. It displays the implementation of the algorithm introduced in Section 6.3, step-by-step. The implementation shown below has been implemented using Knot (Chapter 5), and using the Python Pandas library for data analysis [1].

The purpose of this Appendix is to explicit the algorithm's implementation, for reader's reference.

Introducing Array Programming

The code displayed in this Appendix makes heavy use of Python Pandas for data analysis. Most of this code follows *Array Programming*, where operations are applied to an entire vector or matrix. These operations result in extremely concise and efficient programs, but may be uneasy to understand. One may read the Pandas Introduction Page [2] for a better understanding.

B.1 Implementation

```
events = events.reset_index() \
               .set_index(['house', 'sensor', 'date']) \
               .sort()
```

Listing B.1: Grouping events by sensor

[1] http://pandas.pydata.org
[2] http://pandas.pydata.org/pandas-docs/stable/dsintro.html

```python
def _insert_missing_off_and_resample(x):
    x = x.reset_index()
    # Manually add Off after 60 seconds without On,
    #   To prevent sensor failures
    unmatched_on = x[(x.value == "On") &
                     (x.date.shift(-1) - x.date >
                        timedelta(seconds=70))]
    new_offs = pd.DataFrame({
        'date': unmatched_on.date + timedelta(minutes=1),
        'house': unmatched_on.house,
        'sensor': unmatched_on.sensor,
        'value': 'Off'})
    x = pd.concat([x, new_offs]) \
             .set_index(['date']) \
             .sort_index()
    # Resample the events, with Last Observation Carried Forward
    return x.resample('10S', how='first', fill_method='pad')

# Add missing off and resample for each sensor of each house
events = events.groupby(level='house') \
      .apply(lambda y: y.groupby(level='sensor') \
                  .apply(_insert_missing_off_and_resample))
events = events.drop(['house', 'sensor'], axis=1) \
      .reset_index()
# Create one column per sensor
events = events.pivot(index='date', columns='sensor',
                      values='value')

# Cleaning up the dataframe
events['house'] = "J"
events = events.reset_index() \
               .set_index(['house', 'date'])
```

Listing B.2: Resampling events to uniform frequency, per sensor

```
# Replace every On by Movement, fill-in the blanks
events = events.replace({'On': 'Movement'}) \
               .fillna(method='pad')
# The remaining blanks are before any signal, they are Unknown
events = events.fillna('Unknown')
# If another room is having Movement at the same time,
#   we set it the room presence as Absent
events[events.apply(lambda x: ~x.isin(['Movement', 'Unknown'])
                    & (events == 'Movement').any(1))
      ] = 'Absent'
# For Off signals that are consecutive to a movement
#   Without any other movement elsewhere
#   We set the room presence as Immobile
events[events == 'Off'] = np.nan
events[(events.copy().fillna(method='pad') == 'Movement')
       & pd.isnull(events)
      ] = 'Immobile'
events = events.fillna(method='pad', inplace=True)
```

Listing B.3: Inferring presence in rooms

```
# If there is more than one room with presence,
#   there is a multiuser
events['Multiuser'] = (events.isin(['Movement', 'Immobile']) \
                        .T \
                        .sum()) > 1
```

Listing B.4: Infering multiuser situations

```
# When no multiuser
#   We select the active room at each time
location = events[~events['Multiuser']] \
    .T \
    .isin(['Movement', 'Immobile']) \
    .idxmax() \
# When there is a multiuser,
#   we set the current Location as Multiuser
location = location.combine_first(
        events['Multiuser'][events['Multiuser']]) \
    .replace({True: "Multiuser"})

location = pd.DataFrame(events, columns=['Location'])
```

Listing B.5: Concluding on the resident's location

```
# Filtering consecutive identical rows
location = location.loc[(location.Location.shift()
                         != location.Location)]

def _make_until_column(x):
    x = x.reset_index(level='date')
    x['Until'] = x.date.shift(-1)
    return x.set_index('date', append=True)

# Make a Until column from the next location's date
location = location.groupby(level='house') \
               .apply(_make_until_column) \
               .location.reindex_axis(['Until', 'Location'],
                                      axis=1)
```

Listing B.6: Reducing the location list

Bibliography

References

[1] Mounir Mokhtari, Romain Endelin, Hamdi Aloulou, and Thibaut Tiberghien. "Measuring the impact of ICTs on the quality of life of ageing people with mild dementia". In: *Smart Homes and Health Telematics*. Springer, 2015, pp. 103–109 (cit. on pp. 4, 50, 53, 144).

[2] Anders Wimo and Martin James Prince. *World Alzheimer Report 2010: the global economic impact of dementia*. Alzheimer's Disease International, 2010 (cit. on p. 7).

[3] Jack Diamond. *A report on Alzheimer's disease and current research*. Alzheimer Society of Canada, 2006 (cit. on p. 7).

[4] Barry Reisberg, Steven H Ferris, Mony J de Leon, and Thomas Crook. "The Global Deterioration Scale for assessment of primary degenerative dementia." In: *The American journal of psychiatry* (1982) (cit. on p. 7).

[5] CP Ferri, R Sousa, E Albanese, WS Ribeiro, and M Honyashiki. "World Alzheimer Report 2009-Executive Summary". In: *Edited by: Prince M, Jackson J. London: AlzheimerLs Disease International* (2009), pp. 1–22 (cit. on p. 7).

[6] Ron Brookmeyer, Elizabeth Johnson, Kathryn Ziegler-Graham, and H Michael Arrighi. "Forecasting the global burden of Alzheimer's disease". In: *Alzheimer's & dementia* 3.3 (2007), pp. 186–191 (cit. on p. 7).

[7] Ada-Helen Bayer, Leon Harper, and Mathew Greenwald. *Fixing to stay: A national survey of housing and home modification issues*. American Association of Retired Persons, 2000 (cit. on p. 8).

[8] Lisa Jeannotte, Margaret Jane Moore, et al. *The State of aging and health in America 2007*. Merck Company Foundation, 2007 (cit. on p. 8).

[9] Robert N Butler. "Why survive? Being old in America." In: (1975) (cit. on p. 8).

[10] Carole Haber. *Nursing Homes: History*. URL: `http://medicine.jrank.org/pages/1243/Nursing-Homes-History.html` (visited on 07/08/2015) (cit. on p. 8).

[11] National Institute of Aging. *Global Health and Aging*. URL: `https://www.nia.nih.gov/research/publication/global-health-and-aging/changing-role-family` (visited on 12/24/2015) (cit. on p. 8).

[12] K Jongenelis, AM Pot, AMH Eisses, ATF Beekman, H Kluiter, and MW Ribbe. "Prevalence and risk indicators of depression in elderly nursing home patients: the AGED study". In: *Journal of affective disorders* 83.2 (2004), pp. 135–142 (cit. on p. 8).

[13] Irène Sipos. “Evolution des différentes fonctions en maison de retraite”. In: (2003). In French (cit. on p. 8).

[14] KPMG. *Observatoire des EHPAD*. In French. 2012. URL: http://www.kpmg.com/FR/fr/IssuesAndInsights/ArticlesPublications/Documents/Observatoire-des-EPHAD-2012.pdf (visited on 12/24/2015) (cit. on p. 9).

[15] La Tribune Jean-Christophe Chanut. *La pension moyenne de retraite s'élève à 1.306 euros bruts mensuels*. In French. 2015. URL: http://www.latribune.fr/economie/france/la-pension-moyenne-de-retraite-s-eleve-a-1-306-euros-bruts-mensuels-475756.html (visited on 12/24/2015) (cit. on p. 9).

[16] La Maison de l'Autonomie Daniel CARRE. *Personnes agées. Aidants et maintien à domicile*. In French. 2014. URL: http://www.lamaisondelautonomie.com/aidants-et-maintien-a-domicile/ (visited on 12/24/2015) (cit. on p. 9).

[17] Hélène Kerhervé. “Du domicile à l'institution: évaluation du stress et du soutien par les conjoints et les enfants accompagnant un proche atteint de maladie d'Alzheimer”. PhD thesis. Paris 10, 2010 (cit. on p. 9).

[18] Mark Weiser. “The computer for the 21st century”. In: *Scientific american* 265.3 (1991), pp. 94–104 (cit. on pp. 10, 11).

[19] Mark Weiser. *Ubiquitous Computing*. URL: http://ubiq.com/weiser/UbiHome.html (visited on 07/08/2015) (cit. on p. 10).

[20] Mahadev Satyanarayanan. “Pervasive computing: Vision and challenges”. In: *Personal Communications, IEEE* 8.4 (2001), pp. 10–17 (cit. on p. 10).

[21] Dieter Uckelmann, Mark Harrison, and Florian Michahelles. *Architecting the internet of things*. Springer Science & Business Media, 2011 (cit. on p. 11).

[22] Leo Sun. *International Business Machines Corp. Invests 3$ Billion in the Internet of Things*. 2015. URL: http://www.fool.com/investing/general/2015/04/02/international-business-machines-corp-invests-3-bil.aspx (visited on 07/03/2015) (cit. on p. 12).

[23] Cisco Systems. *The Internet of Things (IoT)*. URL: http://www.cisco.com/web/solutions/trends/iot/overview.html (visited on 07/03/2015) (cit. on p. 12).

[24] Joseph F Coughlin, Lisa A D'Ambrosio, Bryan Reimer, and Michelle R Pratt. “Older adult perceptions of smart home technologies: implications for research, policy & market innovations in healthcare”. In: *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*. IEEE. 2007, pp. 1810–1815 (cit. on p. 13).

[25] Emile Aarts and Reiner Wichert. *Ambient intelligence*. Springer, 2009 (cit. on p. 14).

[26] Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. “A review of smart homes—Present state and future challenges”. In: *Computer methods and programs in biomedicine* 91.1 (2008), pp. 55–81 (cit. on p. 15).

[27] Anind K Dey. "Understanding and using context". In: *Personal and ubiquitous computing* 5.1 (2001), pp. 4–7 (cit. on pp. 18, 37).

[28] Juan Ye, Simon Dobson, and Susan McKeever. "Situation identification techniques in pervasive computing: A review". In: *Pervasive and mobile computing* 8.1 (2012), pp. 36–66 (cit. on pp. 18, 40, 44).

[29] Mounir Mokhtari, Hamdi Aloulou, Thibaut Tiberghien, Jit Biswas, Daniel Racoceanu, and Philip Yap. "New trends to support independence in persons with mild dementia–a mini-review". In: *Gerontology* 58.6 (2012), pp. 554–563 (cit. on p. 19).

[30] Hamdi Aloulou. "Framework for ambient assistive living: handling dynamism and uncertainty in real time semantic services provisioning". PhD thesis. Evry, Institut national des télécommunications, 2014 (cit. on pp. 19, 83, 156).

[31] Thibaut Tiberghien. "Strategies for context reasoning in assistive livings for the elderly". PhD thesis. Evry, Institut national des télécommunications, 2013 (cit. on pp. 19, 81, 111, 121, 156).

[32] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. "N3logic: A logical framework for the world wide web". In: *Theory and Practice of Logic Programming* 8.03 (2008), pp. 249–269 (cit. on pp. 19, 42).

[33] Deborah L McGuinness, Frank Van Harmelen, et al. "OWL web ontology language overview". In: *W3C recommendation* 10.10 (2004), p. 2004 (cit. on pp. 19, 42).

[34] Jos De Roo. "Euler proof mechanism". In: *Internet: http://www. agfa. com/w3c/euler* 40 (2002) (cit. on pp. 19, 42, 111, 156).

[35] Stefano Federici and Marcia Scherer. *Assistive technology assessment handbook*. CRC Press, 2012 (cit. on p. 25).

[36] James R Knickman and Emily K Snell. "The 2030 problem: caring for aging baby boomers". In: *Health services research* 37.4 (2002), pp. 849–884 (cit. on p. 25).

[37] K Spitalewsky, J Rochon, M Ganzinger, P Knaup, et al. "Potential and requirements of IT for ambient assisted living technologies". In: *Methods Inf Med* 52.3 (2013), pp. 231–238 (cit. on p. 25).

[38] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. "A survey on ambient intelligence in healthcare". In: *Proceedings of the IEEE* 101.12 (2013), pp. 2470–2494 (cit. on pp. 25, 28, 34).

[39] Tim Brugger. *3 Stocks to Invest in The Internet of Things*. 2015. URL: `http://www.fool.com/investing/general/2015/04/12/3-stocks-to-invest-in-the-internet-of-things.aspx` (visited on 07/03/2015) (cit. on p. 26).

[40] Ben Popper. *Quirky just announced seven new products, a micro-factory, custom sensor kits, and the death of the thermostat*. 2014. URL: `http://www.theverge.com/2014/11/11/7193765/quirky-ge-wink-uniq-spotter-norm-factory` (visited on 07/03/2015) (cit. on p. 27).

[41] Thomas Kleinberger, Martin Becker, Eric Ras, Andreas Holzinger, and Paul Müller. "Ambient intelligence in assisted living: enable elderly people to handle future interfaces". In: *Universal access in human-computer interaction. Ambient interaction*. Springer, 2007, pp. 103–112 (cit. on p. 28).

[42] Alexandros Mourouzis, Asterios Leonidis, Michalis Foukarakis, Margherita Antona, and Nicos Maglaveras. "A novel design approach for multi-device adaptable user interfaces: concepts, methods and examples". In: *Universal Access in Human-Computer Interaction. Design for All and eInclusion*. Springer, 2011, pp. 400–409 (cit. on p. 28).

[43] Aquitaine Numérique. *Le projet Sacha dans le secteur de la sante*. In French. 2011. URL: `http://numerique.aquitaine.fr/Le-projet-Sacha-dans-le-secteur-de` (visited on 07/03/2015) (cit. on p. 30).

[44] Cyberdocteur. *E-Celcius, la pilule-thermomètre connectée*. In French. 2015. URL: `http://www.medsyn.fr/perso/g.perrin/joomla/index.php/profession-mg/41-generaliste/346-e-celcius-la-pilule-thermometre-connectee` (visited on 07/03/2015) (cit. on p. 30).

[45] T Adlam and Roger D Orpwood. "Taking the Gloucester Smart House from the laboratory to the living room". In: *Proceedings of 3rd International Workshop on Ubiquitous Computing for Pervasive Healthcare Applications*. 2004 (cit. on p. 30).

[46] Alex Mihailidis, Brent Carmichael, and Jennifer Boger. "The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home". In: *Information Technology in Biomedicine, IEEE Transactions on* 8.3 (2004), pp. 238–247 (cit. on p. 30).

[47] Martha E Pollack, Laura Brown, Dirk Colbry, Colleen E McCarthy, Cheryl Orosz, Bart Peintner, Sailesh Ramakrishnan, and Ioannis Tsamardinos. "Autominder: An intelligent cognitive orthotic system for people with memory impairment". In: *Robotics and Autonomous Systems* 44.3 (2003), pp. 273–282 (cit. on p. 30).

[48] Cathy Bodine, Sumi Helal, Tao Gu, and Mounir Mokhtari. *Smart Homes and Health Telematics: 12th International Conference, ICOST 2014, Denver, CO, USA, June 25-27, 2014, Revised Selected Papers*. Vol. 8456. Springer, 2015 (cit. on p. 30).

[49] Antoine Geissbühler, Jacques Demongeot, Mounir Mokhtari, Bessam Abdulrazak, and Hamdi Aloulou. *Inclusive Smart Cities and e-Health: 13th International Conference on Smart Homes and Health Telematics, ICOST 2015, Geneva, Switzerland, June 10-12, 2015, Proceedings*. Vol. 9102. Springer, 2015 (cit. on p. 30).

[50] Joaquim Bellmunt, Thibaut Tiberghien, Mounir Mokhtari, Hamdi Aloulou, and Romain Endelin. "Technical Challenges Towards an AAL Large Scale Deployment". In: *Inclusive Smart Cities and e-Health*. Springer, 2015, pp. 3–14 (cit. on pp. 31, 102).

[51] Mohammad-Reza Tazari, Francesco Furfari, Álvaro Fides-Valero, Sten Hanke, Oliver Höftberger, Dionisis Kehagias, Miran Mosmondor, Reiner Wichert, and Peter Wolf. “The universAAL Reference Model for AAL.” In: *Handbook of Ambient Assisted Living* 11 (2012), pp. 610–625 (cit. on p. 31).

[52] Marilyn J Rantz, Marjorie Skubic, Richelle J Koopman, Linsey Phillips, Gregory L Alexander, Steven J Miller, and Rainer Dane Guevara. “Using sensor networks to detect urinary tract infections in older adults”. In: *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on.* IEEE. 2011, pp. 142–149 (cit. on p. 31).

[53] Andrew Sixsmith, Sonja Meuller, Felicitas Lull, Michael Klein, Ilse Bierhoff, Sarah Delaney, and Robert Savage. “SOPRANO–an ambient assisted living system for supporting older people at home”. In: *Ambient Assistive Health and Wellness Management in the Heart of the City.* Springer, 2009, pp. 233–236 (cit. on p. 31).

[54] Peter Wolf, Andreas Schmidt, Javier Parada Otte, Michael Klein, Sebastian Rollwage, Birgitta König-Ries, Torsten Dettborn, and Aygul Gabdulkhakova. “openAAL-the open source middleware for ambient-assisted living (AAL)”. In: *AALIANCE conference, Malaga, Spain.* 2010, pp. 1–5 (cit. on p. 31).

[55] Parisa Rashidi and Diane J Cook. “Keeping the resident in the loop: Adapting the smart home to the user”. In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 39.5 (2009), pp. 949–959 (cit. on pp. 31, 44, 45).

[56] Adriana M Adami, Misha Pavel, Tamara L Hayes, and Clifford M Singer. “Detection of movement in bed using unobtrusive load cell sensors”. In: *Information Technology in Biomedicine, IEEE Transactions on* 14.2 (2010), pp. 481–490 (cit. on p. 31).

[57] Bruno Bouchard, Sylvain Giroux, and Abdenour Bouzouane. “A keyhole plan recognition model for Alzheimer’s patients: first results”. In: *Applied Artificial Intelligence* 21.7 (2007), pp. 623–658 (cit. on p. 31).

[58] Markus Nick and Martin Becker. “A hybrid approach to intelligent living assistance”. In: *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on.* IEEE. 2007, pp. 283–289 (cit. on p. 31).

[59] Maged NK Boulos, Artur Rocha, Angelo Martins, Manuel E Vicente, Armin Bolz, Robert Feld, Igor Tchoudovski, Martin Braecklein, John Nelson, Gearóid Ó Laighin, et al. “CAALYX: a new generation of location-based services in healthcare”. In: *International Journal of Health Geographics* 6.1 (2007), p. 9 (cit. on p. 31).

[60] Toshiyo Tamura, Atsushi Kawarada, Masayuki Nambu, Akira Tsukada, Kazuo Sasaki, and Ken-Ichi Yamakoshi. “E-healthcare at an experimental welfare techno house in Japan”. In: *The open medical informatics journal* 1 (2007), p. 1 (cit. on p. 31).

[61] Gael LeBellego, Norbert Noury, Gilles Viron, Mireille Mousseau, and Jacques Demong. “A model for the measurement of patient activity in a hospital suite”. In: *Information Technology in Biomedicine, IEEE Transactions on* 10.1 (2006), pp. 92–99 (cit. on p. 31).

[62] Anthony Wood, Gilles Virone, Thao Doan, Quihua Cao, Leo Selavo, Yafeng Wu, L Fang, Zhimin He, Shan Lin, and Jack Stankovic. “ALARM-NET: Wireless sensor networks for assisted-living and residential monitoring”. In: *University of Virginia Computer Science Department Technical Report* 2 (2006) (cit. on p. 31).

[63] Anthony D Wood, John Stankovic, Gilles Virone, Leo Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, Radu Stoleru, et al. “Context-aware wireless sensor networks for assisted living and residential monitoring”. In: *Network, IEEE* 22.4 (2008), pp. 26–33 (cit. on p. 31).

[64] G Michael Youngblood, Diane J Cook, and Lawrence B Holder. “Managing adaptive versatile environments”. In: *Pervasive and Mobile Computing* 1.4 (2005), pp. 373–403 (cit. on p. 31).

[65] Sumi Helal, William Mann, Jeffrey King, Youssef Kaddoura, Erwin Jansen, et al. “The gator tech smart house: A programmable pervasive space”. In: *Computer* 38.3 (2005), pp. 50–60 (cit. on pp. 31, 162).

[66] Marie Chan, Eric Campo, and Daniel Estève. “Assessment of activity of elderly people using a home monitoring system”. In: *International Journal of Rehabilitation Research* 28.1 (2005), pp. 69–76 (cit. on p. 31).

[67] Tatsuya Yamazaki. “Ubiquitous home: real-life testbed for home context-aware service”. In: *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2005. Tridentcom 2005. First International Conference on*. IEEE. 2005, pp. 54–59 (cit. on p. 31).

[68] Mark Perry, Alan Dowdall, Lorna Lines, and Kate Hone. “Multimodal and ubiquitous computing systems: Supporting independent-living older users”. In: *Information Technology in Biomedicine, IEEE Transactions on* 8.3 (2004), pp. 258–270 (cit. on p. 31).

[69] Yoshifumi Nishida, Toshio Hori, Takashi Suehiro, and Shigeoki Hirai. “Sensorized environment for self-communication based on observation of daily human behavior”. In: *Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*. Vol. 2. IEEE. 2000, pp. 1364–1372 (cit. on p. 31).

[70] Gareth Williams, Kevin Doughty, David Bradley, et al. “A systems approach to achieving CarerNet-an integrated and intelligent telecare system”. In: *Information Technology in Biomedicine, IEEE Transactions on* 2.1 (1998), pp. 1–9 (cit. on p. 31).

[71] Dadong Wan. “Magic medicine cabinet: A situated portal for consumer healthcare”. In: *Handheld and Ubiquitous Computing*. Springer. 1999, pp. 352–355 (cit. on p. 30).

[72] Michael D Murray. ". Automated Medication Dispensing Devices". In: *Making Health Care Safer: A Critical Analysis of Patient Safety Practices* 111 (2001) (cit. on p. 30).

[73] Frank Siegemund and Christian Florkemeier. "Interaction in pervasive computing settings using Bluetooth-enabled active tags and passive RFID technology together with mobile phones". In: *Pervasive Computing and Communications, 2003.(PerCom 2003). Proceedings of the First IEEE International Conference on.* IEEE. 2003, pp. 378–387 (cit. on p. 30).

[74] Chris Nugent, Dewar Finlay, Richard Davies, Maurice Mulvenna, Jonathan Wallace, Cristiano Paggetti, Elena Tamburini, and Norman Black. "The next generation of mobile medication management solutions". In: *International journal of electronic healthcare* 3.1 (2006), pp. 7–31 (cit. on p. 30).

[75] Pei-Hsuan Tsai, CY Yu, MY Wang, JK Zao, HC Yeh, CS Shih, and JWS Liu. "iMAT: Intelligent medication administration tools". In: *e-Health Networking Applications and Services (Healthcom), 2010 12th IEEE International Conference on.* IEEE. 2010, pp. 308–315 (cit. on p. 30).

[76] Upkar Varshney. "Wireless medication management system: Design and performance evaluation". In: *Wireless Telecommunications Symposium (WTS), 2011.* IEEE. 2011, pp. 1–8 (cit. on p. 30).

[77] Liana Kiff, Karen Haigh, and Xianghong Sun. "Mobility Monitoring with the Independent LifeStyle Assistant(ILSA)". In: *International conference on aging, disability and independence (ICADI).* 2003 (cit. on p. 30).

[78] Mark Buckland, Barbara Frost, and Andrew Reeves. "Liverpool Telecare Pilot: telecare as an information tool". In: *Informatics in primary care* 14.3 (2006), pp. 191–196 (cit. on p. 30).

[79] Toshiro Suzuki, Sumio Murase, Tomoyuki Tanaka, and Takako Okazawa. "New approach for the early detection of dementia by recording in-house activities". In: *Telemedicine and e-Health* 13.1 (2007), pp. 41–44 (cit. on p. 30).

[80] Tamara L Hayes, Francena Abendroth, Andre Adami, Misha Pavel, Tracy A Zitzelberger, and Jeffrey A Kaye. "Unobtrusive assessment of activity patterns associated with mild cognitive impairment". In: *Alzheimer's & Dementia* 4.6 (2008), pp. 395–405 (cit. on p. 30).

[81] Céline Franco, Jacques Demongeot, Christophe Villemazet, and Nicolas Vuillerme. "Behavioral telemonitoring of the elderly at home: Detection of nycthemeral rhythms drifts from location data". In: *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on.* IEEE. 2010, pp. 759–766 (cit. on p. 30).

[82] Mladen Milovsevic, Michael T Shrove, and Emil Jovanov. "Applications of smartphones for ubiquitous health monitoring and wellbeing management". In: *JITA-Journal of Information Technology and Applications (Banja Luka)-APEIRON* 1.1 (2011) (cit. on p. 30).

[83] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. “Internet of things for smart cities”. In: *Internet of Things Journal, IEEE* 1.1 (2014), pp. 22–32 (cit. on p. 30).

[84] John Stankovic et al. “Research directions for the internet of things”. In: *Internet of Things Journal, IEEE* 1.1 (2014), pp. 3–9 (cit. on p. 30).

[85] Saisakul Chernbumroong, Shuang Cang, Anthony Atkins, and Hongnian Yu. “Elderly activities recognition and classification for applications in assisted living”. In: *Expert Systems with Applications* 40.5 (2013), pp. 1662–1674 (cit. on pp. 32, 50).

[86] Robert Beringer, Andrew Sixsmith, Michael Campo, Julie Brown, and Rose McCloskey. “The “acceptance” of ambient assisted living: Developing an alternate methodology to this limited research lens”. In: *Toward Useful Services for Elderly and People with Disabilities*. Springer, 2011, pp. 161–167 (cit. on p. 32).

[87] Martina Ziefle, Carsten Röcker, and Andreas Holzinger. “Medical technology in smart homes: exploring the user’s perspective on privacy, intimacy and trust”. In: *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*. IEEE. 2011, pp. 410–415 (cit. on p. 32).

[88] Ken Ducatel, Marc Bogdanowicz, Fabiana Scapolo, Jos Leijten, and Jean-Claude Burgelman. *Scenarios for ambient intelligence in 2010*. Office for official publications of the European Communities, 2001 (cit. on p. 32).

[89] Alejandro Ibarz, Jorge L Falcó, Esteban Vaquerizo, Luis Lain, Jose Ignacio Artigas, and Armando Roy. “MonAMI: Economic impact of mainstream AAL open platform in home care”. In: *Ambient Assisted Living and Home Care*. Springer, 2012, pp. 127–134 (cit. on p. 32).

[90] Gillian R Hayes, Erika Shehan Poole, Giovanni Iachello, Shwetak N Patel, Mikhail Grimes, Gregory D Abowd, and Khai N Truong. “Physical, social, and experiential knowledge in pervasive computing environments”. In: *Pervasive Computing, IEEE* 6.4 (2007), pp. 56–63 (cit. on p. 32).

[91] Peter Rothenpieler, Claudia Becker, and Stefan Fischer. “Privacy concerns in a remote monitoring and social networking platform for assisted living”. In: *Privacy and Identity Management for Life*. Springer, 2011, pp. 219–230 (cit. on p. 32).

[92] Harold Van Heerde, Nicolas Anciaux, Ling Feng, and Peter MG Apers. “Balancing smartness and privacy for the Ambient Intelligence”. In: *Smart Sensing and Context*. Springer, 2006, pp. 255–258 (cit. on p. 32).

[93] Marc Langheinrich. “Privacy by design—principles of privacy-aware ubiquitous systems”. In: *Ubicomp 2001: Ubiquitous Computing*. Springer. 2001, pp. 273–291 (cit. on p. 32).

[94] William Lowrance. “Learning from experience: privacy and the secondary use of data in health research”. In: *Journal of health services research & policy* 8.suppl 1 (2003), pp. 2–7 (cit. on p. 32).

[95] George Yee, Larry Korba, and Ronggong Song. “Ensuring privacy for e-health services”. In: *Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on*. IEEE. 2006, 8–pp (cit. on p. 32).

[96] David Wright, Serge Gutwirth, Michael Friedewald, Elena Vildjiounaite, and Yves Punie. *Safeguards in a world of ambient intelligence*. Vol. 1. Springer Science & Business Media, 2008 (cit. on p. 32).

[97] Jürgen Bohn, Vlad Coroamă, Marc Langheinrich, Friedemann Mattern, and Michael Rohs. “Living in a world of smart everyday objects—Social, economic, and ethical implications”. In: *Human and Ecological Risk Assessment* 10.5 (2004), pp. 763–785 (cit. on p. 32).

[98] Charalampos Doukas, Ilias Maglogiannis, Vassiliki Koufi, Flora Malamateniou, and George Vassilacopoulos. “Enabling data protection through PKI encryption in IoT M-Health devices”. In: *Bioinformatics & Bioengineering (BIBE), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 25–29 (cit. on p. 32).

[99] Alessandra De Paola. “A Cognitive Architecture for Ambient Intelligence”. PhD thesis. University of Palermo, 2011 (cit. on p. 32).

[100] Alessandra Flammini, Paolo Ferrari, Daniele Marioli, Emiliano Sisinni, and Andrea Taroni. “Wired and wireless sensor networks for industrial applications”. In: *Microelectronics Journal* 40.9 (2009), pp. 1322–1336 (cit. on p. 34).

[101] Luca Benini, Elisabetta Farella, and Carlotta Guiducci. “Wireless sensor networks: Enabling technology for ambient intelligence”. In: *Microelectronics journal* 37.12 (2006), pp. 1639–1649 (cit. on p. 34).

[102] Ian F Akyildiz, Xudong Wang, and Weilin Wang. “Wireless mesh networks: a survey”. In: *Computer networks* 47.4 (2005), pp. 445–487 (cit. on p. 35).

[103] Jan Sliwa and Emmanuel Benoist. “Wireless sensor and actor networks: E-health, E-science, E-decisions”. In: *Mobile and Wireless Networking (iCOST), 2011 International Conference on Selected Topics in*. IEEE. 2011, pp. 1–6 (cit. on p. 35).

[104] Flávia C Delicato, Lidia Fuentes, Nadia Gámez, and Paulo F Pires. “Variabilities of wireless and actuators sensor network middleware for ambient assisted living”. In: *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. Springer, 2009, pp. 851–858 (cit. on p. 35).

[105] Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C Leung. “Body area networks: A survey”. In: *Mobile networks and applications* 16.2 (2011), pp. 171–193 (cit. on p. 35).

[106] Benoît Latré, Bart Braem, Ingrid Moerman, Chris Blondia, and Piet Demeester. “A survey on wireless body area networks”. In: *Wireless Networks* 17.1 (2011), pp. 1–18 (cit. on p. 35).

[107] Cheryl D Metcalf, SRl Collie, AW Cranny, Georgie Hallett, Christopher James, Jo Adams, PH Chappell, NM White, and JH Burridge. “Fabric-based strain sensors for measuring movement in wearable telemonitoring applications”. In: (2009) (cit. on p. 35).

[108] Allen Y Yang, Roozbeh Jafari, S Shankar Sastry, and Ruzena Bajcsy. “Distributed recognition of human actions using wearable motion sensor networks”. In: *Journal of Ambient Intelligence and Smart Environments* 1.2 (2009), pp. 103–115 (cit. on p. 35).

[109] Eric J Pauwels, Albert A Salah, and Romain Tavenard. “Sensor networks for ambient intelligence”. In: *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*. IEEE. 2007, pp. 13–16 (cit. on p. 35).

[110] Parisa Rashidi and Alex Mihailidis. “A survey on ambient-assisted living tools for older adults”. In: *Biomedical and Health Informatics, IEEE Journal of* 17.3 (2013), pp. 579–590 (cit. on p. 35).

[111] Masayuki Nambu, Kazuki Nakajima, Makoto Noshiro, and Toshiyo Tamura. “An algorithm for the automatic detection of health conditions”. In: *Engineering in Medicine and Biology Magazine, IEEE* 24.4 (2005), pp. 38–42 (cit. on p. 35).

[112] Hisanori Andoh, Takayuki Ishikawa, Keita Kobayashi, Kazuyuki Kobayashi, Kajiro Watanabe, and Testuo Nakamura. “Home health monitoring system in the sleep”. In: *SICE Annual Conference in Fukui*. 2003, pp. 1223–1226 (cit. on p. 35).

[113] Ivo Iliev and Ivan Dotsinsky. “Assisted Living Systems for elderly and disabled people: A short review”. In: *Int. J. Bioautom* 15 (2011), pp. 131–139 (cit. on p. 36).

[114] Pablo Oliveira Antonino, Daniel Schneider, Cristian Hofmann, and Elisa Yumi Nakagawa. “Evaluation of AAL platforms according to architecture-based quality attributes”. In: *Ambient Intelligence*. Springer, 2011, pp. 264–274 (cit. on p. 36).

[115] Miguel Damas, Hector Pomares, Sergio Gonzalez, Alberto Olivares, and Ignacio Rojas. “Ambient assisted living devices interoperability based on OSGi and the X73 standard”. In: *TELEMEDICINE and e-HEALTH* 19.1 (2013), pp. 54–60 (cit. on p. 36).

[116] Julio Abascal, Borja Bonail, Luis Gardeazabal, Alberto Lafuente, and Zigor Salvador. “Managing intelligent services for people with disabilities and elderly people”. In: *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer, 2009, pp. 623–630 (cit. on p. 36).

[117] Juan M Corchado, Javier Bajo, Dante Tapia, Ajith Abraham, et al. “Using heterogeneous wireless sensor networks in a telemonitoring system for healthcare”. In: *Information Technology in Biomedicine, IEEE Transactions on* 14.2 (2010), pp. 234–240 (cit. on p. 36).

[118] Marius Mikalsen, Sten Hanke, Thomas Fuxreiter, Ståle Walderhaug, Leendert WM Wienhofen, and N Trondheim. “Interoperability services in the MPOWER Ambient Assisted Living platform.” In: *MIE*. Vol. 150. 2009, pp. 366–370 (cit. on p. 36).

[119] Hamdi Aloulou, Mounir Mokhtari, Thibaut Tiberghien, Jit Biswas, and Lin Jin Hong Kenneth. “A semantic plug&play based framework for ambient assisted living”. In: *Impact Analysis of solutions for chronic disease Prevention and Management*. Springer, 2012, pp. 165–172 (cit. on p. 36).

[120] Andreas Hein, Marco Eichelberg, Oliver Nee, Arne Schulz, Axel Helmer, and Myriam Lipprandt. "A service oriented platform for health services and ambient assisted living". In: *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on.* IEEE. 2009, pp. 531–537 (cit. on p. 36).

[121] Stefan Wagner and Claus Nielsen. "OpenCare project: An open, flexible and easily extendible infrastructure for pervasive healthcare assisted living solutions". In: *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on.* IEEE. 2009, pp. 1–10 (cit. on p. 36).

[122] Mario Vega-Barbas, Diego Casado-Mansilla, Miguel Valero, Diego López-de-Ipina, José Bravo, Francisco Flórez, et al. "Smart spaces and smart objects interoperability architecture (S3OiA)". In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on.* IEEE. 2012, pp. 725–730 (cit. on p. 36).

[123] Javier Andréu Pérez, Juan Antonio Álvarez, Alejandro Fernández-Montes, and Juan Antonio Ortega. "Service-oriented device integration for ubiquitous ambient assisted living environments". In: *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living.* Springer, 2009, pp. 843–850 (cit. on p. 36).

[124] Ji Eun Kim, George Boulos, John Yackovich, Tassilo Barth, Christian Beckel, and Daniel Mosse. "Seamless integration of heterogeneous devices and access control in smart homes". In: *Intelligent Environments (IE), 2012 8th International Conference on.* IEEE. 2012, pp. 206–213 (cit. on p. 36).

[125] Stuart Russell and Peter Norvig. "Artificial intelligence: a modern approach". In: (1995) (cit. on pp. 36, 123).

[126] Nicola Gatti, Francesco Amigoni, and Marco Rolando. "Multiagent technology solutions for planning in ambient intelligence". In: *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 02.* IEEE Computer Society. 2008, pp. 286–289 (cit. on p. 37).

[127] Francesco Amigoni, Nicola Gatti, Carlo Pinciroli, and Manuel Roveri. "What planner for ambient intelligence applications?" In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 35.1 (2005), pp. 7–21 (cit. on p. 37).

[128] Hans W Guesgen and Stephen Marsland. "Spatio-temporal reasoning and context awareness". In: *Handbook of Ambient Intelligence and Smart Environments.* Springer, 2010, pp. 609–634 (cit. on p. 37).

[129] Jon Barwise and John Perry. *The situation underground.* Stanford University Press, 1980 (cit. on p. 37).

[130] Juan Ye, Susan McKeever, Lorcan Coyle, Steve Neely, and Simon Dobson. "Resolving uncertainty in context integration and abstraction: context integration and abstraction". In: *Proceedings of the 5th international conference on Pervasive services*. ACM. 2008, pp. 131–140 (cit. on pp. 38, 44).

[131] Hedda R Schmidtke and Woontack Woo. "Towards ontology-based formal verification methods for context aware systems". In: *Pervasive Computing*. Springer, 2009, pp. 309–326 (cit. on p. 38).

[132] Juan Ye, Lorcan Coyle, Simon Dobson, and Paddy Nixon. "Using situation lattices to model and reason about context". In: *Proceedings of MRC 2007 (coexist with CONTEXT'07)* (2007), pp. 1–12 (cit. on pp. 38, 45).

[133] Iain McCowan, Daniel Gatica-Perez, Samy Bengio, Guillaume Lathoud, Mark Barnard, and Dong Zhang. "Automatic analysis of multimodal group actions in meetings". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.3 (2005), pp. 305–317 (cit. on p. 38).

[134] Driss Choujaa and Naranker Dulay. "Activity inference through sequence alignment". In: *Location and Context Awareness*. Springer, 2009, pp. 19–36 (cit. on p. 38).

[135] Vikramaditya R Jakkula, Aaron S Crandall, and Diane J Cook. "Knowledge discovery in entity based smart environment resident data using temporal relation based data mining". In: *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE. 2007, pp. 625–630 (cit. on p. 38).

[136] Claudio Bettini, Oliver Brdiczka, Karen Henricksen, Jadwiga Indulska, Daniela Nicklas, Anand Ranganathan, and Daniele Riboni. "A survey of context modelling and reasoning techniques". In: *Pervasive and Mobile Computing* 6.2 (2010), pp. 161–180 (cit. on p. 38).

[137] Parisa Rashidi, Diane J Cook, Lawrence B Holder, and Maureen Schmitter-Edgecombe. "Discovering activities to recognize and track in a smart environment". In: *Knowledge and Data Engineering, IEEE Transactions on* 23.4 (2011), pp. 527–539 (cit. on pp. 40, 45, 46).

[138] Karen Henricksen and Jadwiga Indulska. "Developing context-aware pervasive computing applications: Models and approach". In: *Pervasive and mobile computing* 2.1 (2006), pp. 37–64 (cit. on p. 40).

[139] Seng W Loke. "Logic programming for context-aware pervasive computing: Language support, characterizing situations, and integration with the web". In: *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. IEEE Computer Society. 2004, pp. 44–50 (cit. on p. 40).

[140] Seng W Loke. "Incremental awareness and compositionality: A design philosophy for context-aware pervasive systems". In: *Pervasive and Mobile Computing* 6.2 (2010), pp. 239–253 (cit. on p. 40).

[141] Seng W Loke. “Representing and reasoning with situations for context-aware pervasive computing: a logic programming perspective”. In: *The Knowledge Engineering Review* 19.03 (2004), pp. 213–233 (cit. on p. 40).

[142] Mehul Bhatt and Seng Loke. “Modelling dynamic spatial systems in the situation calculus”. In: *Spatial Cognition & Computation* 8.1-2 (2008), pp. 86–130 (cit. on p. 40).

[143] James L Crowley, Joëlle Coutaz, Gaeten Rey, and Patrick Reignier. “Perceptual components for context aware computing”. In: *UbiComp 2002: Ubiquitous Computing*. Springer, 2002, pp. 117–134 (cit. on p. 40).

[144] Anupama Kalyan, Srividya Gopalan, and V Sridhar. “Hybrid context model based on multilevel situation theory and ontology for contact centers”. In: *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on*. IEEE. 2005, pp. 3–7 (cit. on p. 40).

[145] John McCarthy and Sasa Buvac. “Formalizing context (expanded notes)”. In: (1997) (cit. on p. 40).

[146] John McCarthy. “Situation calculus with concurrent events and narrative”. In: (2000) (cit. on p. 40).

[147] Hector Levesque, Fiora Pirri, and Ray Reiter. “Foundations for the situation calculus”. In: *Linköping Electronic Articles in Computer and Information Science* 3.18 (1998) (cit. on p. 40).

[148] R Michael Furr and David C Funder. “Situational similarity and behavioral consistency: Subjective, objective, variable-centered, and person-centered approaches”. In: *Journal of Research in Personality* 38.5 (2004), pp. 421–447 (cit. on p. 40).

[149] James F Allen. “Maintaining knowledge about temporal intervals”. In: *Communications of the ACM* 26.11 (1983), pp. 832–843 (cit. on p. 41).

[150] David A Randell, Zhan Cui, and Anthony G Cohn. “A spatial logic based on regions and connection.” In: *KR* 92 (1992), pp. 165–176 (cit. on p. 41).

[151] Björn Gottfried, Hans W Guesgen, and Sebastian Hübner. “Spatiotemporal reasoning for smart homes”. In: *Designing Smart Homes*. Springer, 2006, pp. 16–34 (cit. on p. 41).

[152] Obinna Anya, Hissam Tawfik, Saad Amin, Atulya Nagar, and Khaled Shaalan. “Context-aware knowledge modelling for decision support in e-health”. In: *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE. 2010, pp. 1–7 (cit. on p. 42).

[153] Marko Luther, Thorsten Liebig, Sebastian Böhm, and Olaf Noppens. “Who the Heck is the Father of Bob?” In: *The Semantic Web: Research and Applications*. Springer, 2009, pp. 66–80 (cit. on p. 42).

[154] Thomas R Gruber. “Toward principles for the design of ontologies used for knowledge sharing?” In: *International journal of human-computer studies* 43.5 (1995), pp. 907–928 (cit. on p. 42).

[155] Zachary Wemlinger and Lawrence Holder. “The cose ontology: Bringing the semantic web to smart environments”. In: *Toward Useful Services for Elderly and People with Disabilities*. Springer, 2011, pp. 205–209 (cit. on p. 42).

[156] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. “Hybrid reasoning in the CARE middleware for context awareness”. In: *International journal of Web engineering and technology* 5.1 (2009), pp. 3–23 (cit. on p. 42).

[157] Harry Chen, Filip Perich, Tim Finin, and Anupam Joshi. “Soupa: Standard ontology for ubiquitous and pervasive applications”. In: *Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on*. IEEE. 2004, pp. 258–267 (cit. on p. 42).

[158] Daqing Zhang, Tao Gu, and Xiaohang Wang. “Enabling context-aware smart home with semantic web technologies”. In: *International Journal of Human-friendly Welfare Robotic Systems* 6.4 (2005), pp. 12–20 (cit. on p. 42).

[159] Ioanna Roussaki, Maria Strimpakou, Nikos Kalatzis, Miltos Anagnostou, and Carsten Pils. “Hybrid context modeling: A location-based scheme using ontologies”. In: *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*. IEEE. 2006, 6–pp (cit. on p. 42).

[160] Paolo Bouquet, Fausto Giunchiglia, Frank van Harmelen, Luciano Serafini, and Heiner Stuckenschmidt. “Contextualizing ontologies”. In: *Web Semantics: Science, Services and Agents on the World Wide Web* 1.4 (2004), pp. 325–343 (cit. on p. 42).

[161] Harry Chen, Tim Finin, and Amupam Joshi. *Semantic web in the context broker architecture*. Tech. rep. DTIC Document, 2005 (cit. on p. 42).

[162] Tao Gu, Xiao Hang Wang, Hung Keng Pung, and Da Qing Zhang. “An ontology-based context model in intelligent environments”. In: *Proceedings of communication networks and distributed systems modeling and simulation conference*. Vol. 2004. 2004, pp. 270–275 (cit. on p. 42).

[163] Anand Ranganathan, Robert E McGrath, Roy H Campbell, and M Dennis Mickunas. “Use of ontologies in a pervasive computing environment”. In: *The Knowledge Engineering Review* 18.03 (2003), pp. 209–220 (cit. on p. 42).

[164] Fabien L Gandon and Norman M Sadeh. “A semantic e-wallet to reconcile privacy and context awareness”. In: *The Semantic Web-ISWC 2003*. Springer, 2003, pp. 385–401 (cit. on p. 42).

[165] Xiao Hang Wang, Da Qing Zhang, Tao Gu, and Hung Keng Pung. “Ontology based context modeling and reasoning using OWL”. In: *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. Ieee. 2004, pp. 18–22 (cit. on p. 42).

[166] Michael Klein, Andreas Schmidt, and Rolf Lauer. "Ontology-centred design of an ambient middleware for assisted living: The case of soprano". In: *Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU), 30th Annual German Conference on Artificial Intelligence (KI 2007), Osnabrück*. 2007 (cit. on p. 42).

[167] Dario Bonino and Fulvio Corno. *Dogont-ontology modeling for intelligent domotic environments*. Springer, 2008 (cit. on p. 42).

[168] Christopher J Matheus, Mieczyslaw M Kokar, and Kenneth Baclawski. "A core ontology for situation awareness". In: *Proceedings of the Sixth International Conference on Information Fusion*. Vol. 1. 2003, pp. 545–552 (cit. on p. 42).

[169] Bernd Mrohs, Marko Luther, Raju Vaidya, Matthias Wagner, Stephan Steglich, Wolfgang Kellerer, and Stefan Arbanowski. "OWL-SF–a distributed semantic service framework". In: *Proc. of the Workshop on Context Awareness for Proactive Systems (CAPS'05), Helsinki*. Citeseer. 2005, pp. 67–77 (cit. on p. 42).

[170] Norbert Weißenberg, Rüdiger Gartmann, and Agnès Voisard. "An ontology-based approach to personalized situation-aware mobile service supply". In: *GeoInformatica* 10.1 (2006), pp. 55–90 (cit. on p. 42).

[171] Alexander Smirnov, Alexey Kashevnik, Tatiana Levashova, Michael Pashkin, and Nikolai Shilov. "Situation modeling in decision support systems". In: *Integration of Knowledge Intensive Multi-Agent Systems, 2007. KIMAS 2007. International Conference on*. IEEE. 2007, pp. 34–39 (cit. on p. 42).

[172] Emmanuel Munguia Tapia, Tanzeem Choudhury, and Matthai Philipose. "Building reliable activity models using hierarchical shrinkage and mined ontology". In: *Pervasive Computing*. Springer, 2006, pp. 17–32 (cit. on p. 42).

[173] Graeme Stevenson, Stephen Knox, Simon Dobson, and Paddy Nixon. "Ontonym: a collection of upper ontologies for developing pervasive systems". In: *Proceedings of the 1st Workshop on Context, Information and Ontologies*. ACM. 2009, p. 9 (cit. on p. 42).

[174] Fatiha Latfi, Bernard Lefebvre, and Céline Descheneaux. "Ontology-Based Management of the Telehealth Smart Home, Dedicated to Elderly in Loss of Cognitive Autonomy." In: *OWLED*. Vol. 258. 2007 (cit. on p. 42).

[175] Naoharu Yamada, Kenji Sakamoto, Goro Kunito, Yoshinori Isoda, Kenichi Yamazaki, and Satoshi Tanaka. "Applying ontology and probabilistic model to human activity recognition from surrounding things". In: *Information and Media Technologies* 2.4 (2007), pp. 1286–1297 (cit. on p. 42).

[176] Stephen S Yau and Junwei Liu. "Hierarchical situation modeling and reasoning for pervasive computing". In: *Software Technologies for Future Embedded and Ubiquitous Systems, 2006 and the 2006 Second International Workshop on Collaborative Computing, Integration, and Assurance. SEUS 2006/WCCIA 2006. The Fourth IEEE Workshop on*. IEEE. 2006, 6–pp (cit. on p. 42).

[177] Ian Horrocks, Peter F Patel-Schneider, and Frank Van Harmelen. "From SHIQ and RDF to OWL: The making of a web ontology language". In: *Web semantics: science, services and agents on the World Wide Web* 1.1 (2003), pp. 7–26 (cit. on p. 42).

[178] Alessandra Agostini, Claudio Bettini, and Daniele Riboni. "Experience report: ontological reasoning for context-aware Internet services". In: (2006) (cit. on p. 42).

[179] Franz Baader. *The description logic handbook: theory, implementation, and applications*. Cambridge university press, 2003 (cit. on p. 42).

[180] Ian Horrocks, Peter F Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, Mike Dean, et al. "SWRL: A semantic web rule language combining OWL and RuleML". In: *W3C Member submission* 21 (2004), p. 79 (cit. on p. 42).

[181] Tarak Chaari, Dejene Ejigu, Frédérique Laforest, and Vasile-Marian Scuturici. "A comprehensive approach to model and use context for adapting applications in pervasive environments". In: *Journal of Systems and Software* 80.12 (2007), pp. 1973–1992 (cit. on p. 42).

[182] Thibaut Tiberghien, Mounir Mokhtari, Hamdi Aloulou, and Jit Biswas. "Semantic reasoning in context-aware assistive environments to support ageing with dementia". In: *The Semantic Web–ISWC 2012*. Springer, 2012, pp. 212–227 (cit. on p. 42).

[183] Umberto Straccia. "Towards a fuzzy description logic for the semantic web (preliminary report)". In: *The Semantic Web: Research and Applications*. Springer, 2005, pp. 167–181 (cit. on p. 43).

[184] Zhongli Ding and Yun Peng. "A probabilistic extension to ontology language OWL". In: *System Sciences, 2004. Proceedings of the 37th Annual Hawaii international conference on*. IEEE. 2004, 10–pp (cit. on pp. 43, 45).

[185] Hui Lei, Daby M Sow, John S Davis II, Guruduth Banavar, and Maria R Ebling. "The design and applications of a context service". In: *ACM SIGMOBILE Mobile Computing and Communications Review* 6.4 (2002), pp. 45–55 (cit. on p. 43).

[186] Anand Ranganathan, Jalal Al-Muhtadi, Shiva Chetan, Roy Campbell, and M Dennis Mickunas. "Middlewhere: a middleware for location awareness in ubiquitous computing applications". In: *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. Springer-Verlag New York, Inc. 2004, pp. 397–416 (cit. on p. 43).

[187] Daniele Riboni and Claudio Bettini. "Context-aware activity recognition through a combination of ontological and statistical reasoning". In: *Ubiquitous Intelligence and Computing*. Springer, 2009, pp. 39–53 (cit. on p. 43).

[188] Arthur P Dempster. "A generalization of Bayesian inference". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1968), pp. 205–247 (cit. on p. 43).

[189] Glenn Shafer et al. *A mathematical theory of evidence*. Vol. 1. Princeton university press Princeton, 1976 (cit. on p. 43).

[190] Xin Hong, Chris Nugent, Maurice Mulvenna, Sally McClean, Bryan Scotney, and Steven Devlin. “Evidential fusion of sensor data for activity recognition in smart homes”. In: *Pervasive and Mobile Computing* 5.3 (2009), pp. 236–252 (cit. on p. 43).

[191] Susan Mckeever, Juan Ye, Lorcan Coyle, Chris Bleakley, and Simon Dobson. “Activity recognition using temporal evidence theory”. In: (2010) (cit. on p. 43).

[192] Daqiang Zhang, Jiannong Cao, Jingyu Zhou, and Minyi Guo. “Extended dempster-shafer theory in context reasoning for ubiquitous computing environments”. In: *Computational Science and Engineering, 2009. CSE’09. International Conference on*. Vol. 2. IEEE. 2009, pp. 205–212 (cit. on p. 43).

[193] Susan McKeever, Juan Ye, Lorcan Coyle, and Simon Dobson. “Using dempster-shafer theory of evidence for situation inference”. In: *Smart Sensing and Context*. Springer, 2009, pp. 149–162 (cit. on p. 43).

[194] Hamdi Aloulou, Mounir Mokhtari, Thibaut Tiberghien, Romain Endelin, and Jit Biswas. “Uncertainty handling in semantic reasoning for accurate context understanding”. In: *Knowledge-Based Systems* 77 (2015), pp. 16–28 (cit. on pp. 43, 50).

[195] Lotfi A Zadeh. “Fuzzy sets”. In: *Information and control* 8.3 (1965), pp. 338–353 (cit. on p. 44).

[196] Pari Delir Haghighi, Shonali Krishnaswamy, Arkady Zaslavsky, and Mohamed Medhat Gaber. “Reasoning about context in uncertain pervasive computing environments”. In: *Smart Sensing and Context*. Springer, 2008, pp. 112–125 (cit. on p. 44).

[197] Christos B Anagnostopoulos, Yiorgos Ntarladimas, and Stathes Hadjiefthymiades. “Situational computing: An innovative architecture with imprecise reasoning”. In: *Journal of Systems and Software* 80.12 (2007), pp. 1993–2014 (cit. on p. 44).

[198] Ronald Fagin, Joseph Y Halpern, and Nimrod Megiddo. “A logic for reasoning about probabilities”. In: *Information and computation* 87.1 (1990), pp. 78–128 (cit. on p. 44).

[199] Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. “A probabilistic room location service for wireless networked environments”. In: *Ubicomp 2001: Ubiquitous Computing*. Springer. 2001, pp. 18–34 (cit. on p. 44).

[200] Anand Ranganathan, Jalal Al-Muhtadi, and Roy H Campbell. “Reasoning about uncertain contexts in pervasive computing environments”. In: *IEEE Pervasive Computing* 2 (2004), pp. 62–70 (cit. on pp. 44, 45).

[201] Waltenegus Dargie. “The role of probabilistic schemes in multisensor context-awareness”. In: *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops’ 07. Fifth Annual IEEE International Conference on*. IEEE. 2007, pp. 27–32 (cit. on p. 44).

[202] Chun-Wei Tsai, Chin-Feng Lai, Ming-Chao Chiang, and Laurence T Yang. "Data mining for internet of things: A survey". In: *Communications Surveys & Tutorials, IEEE* 16.1 (2014), pp. 77–97 (cit. on p. 44).

[203] Marja Ruotsalainen, Timo Ala-Kleemola, and Ari Visa. "GAIS: A method for detecting interleaved sequential patterns from imperfect data". In: *Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on*. IEEE. 2007, pp. 530–534 (cit. on pp. 44, 45).

[204] Oliver Brdiczka, James L Crowley, and Patrick Reignier. "Learning situation models in a smart home". In: *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* 39.1 (2009), pp. 56–63 (cit. on p. 45).

[205] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. *Activity recognition in the home using simple and ubiquitous sensors*. Springer, 2004 (cit. on p. 45).

[206] Tim van Kasteren and Ben Krose. "Bayesian activity recognition in residence for elders". In: (2007) (cit. on p. 45).

[207] Donald J Patterson, Lin Liao, Dieter Fox, and Henry Kautz. "Inferring high-level behavior from low-level sensors". In: *UbiComp 2003: Ubiquitous Computing*. Springer. 2003, pp. 73–89 (cit. on p. 45).

[208] Daniel Lowd and Pedro Domingos. "Naive Bayes models for probability estimation". In: *Proceedings of the 22nd international conference on Machine learning*. ACM. 2005, pp. 529–536 (cit. on p. 45).

[209] Daniel H Wilson and Chris Atkeson. "Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors". In: *Pervasive computing*. Springer, 2005, pp. 62–79 (cit. on pp. 45, 46).

[210] Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. "Learning and inferring transportation routines". In: *Artificial Intelligence* 171.5 (2007), pp. 311–331 (cit. on p. 45).

[211] Diane Cook, Kyle D Feuz, and Narayanan C Krishnan. "Transfer learning for activity recognition: A survey". In: *Knowledge and information systems* 36.3 (2013), pp. 537–556 (cit. on p. 45).

[212] Tao Gu, Hung Keng Pung, Da Qing Zhang, Hung Keng Pung, and Da Qing Zhang. *A bayesian approach for dealing with uncertain contexts*. na, 2004 (cit. on p. 45).

[213] Wegdan Abdelsalam and Yasser Ebrahim. "A Bayesian-networks-based approach for managing uncertainty in location-tracking applications". In: *Electrical and Computer Engineering, 2004. Canadian Conference on*. Vol. 4. IEEE. 2004, pp. 2205–2208 (cit. on p. 45).

[214] Binh An Truong, Young-Koo Lee, and Sung-Young Lee. "Modeling uncertainty in context-aware computing". In: *Computer and Information Science, 2005. Fourth Annual ACIS International Conference on*. IEEE. 2005, pp. 676–681 (cit. on p. 45).

[215] Uwe Maurer, Asim Smailagic, Daniel P Siewiorek, and Michael Deisher. “Activity recognition and monitoring using multiple sensors on different body positions”. In: *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on.* IEEE. 2006, 4–pp (cit. on p. 45).

[216] MJ Mathie, Branko G Celler, Nigel H Lovell, and ACF Coster. “Classification of basic daily movements using a triaxial accelerometer”. In: *Medical and Biological Engineering and Computing* 42.5 (2004), pp. 679–687 (cit. on p. 45).

[217] Dean M Karantonis, Michael R Narayanan, Merryn Mathie, Nigel H Lovell, and Branko G Celler. “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring”. In: *Information Technology in Biomedicine, IEEE Transactions on* 10.1 (2006), pp. 156–167 (cit. on p. 45).

[218] Karen SK Cheung and Douglas Vogel. “Complexity reduction in lattice-based information retrieval”. In: *Information retrieval* 8.2 (2005), pp. 285–299 (cit. on p. 45).

[219] Shwetak N Patel, Thomas Robertson, Julie A Kientz, Matthew S Reynolds, and Gregory D Abowd. “At the flick of a switch: Detecting and classifying unique electrical events on the residential power line”. In: *Lecture Notes in Computer Science* 4717 (2007), pp. 271–288 (cit. on p. 45).

[220] Takayuki Kanda, Dylan F Glas, Masahiro Shiomi, Hiroshi Ishiguro, and Norihiro Hagita. “Who will be the customer?: a social robot that anticipates people’s behavior from their trajectories”. In: *Proceedings of the 10th international conference on Ubiquitous computing.* ACM. 2008, pp. 380–389 (cit. on p. 45).

[221] Anthony Fleury, Michel Vacher, and Norbert Noury. “SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results”. In: *Information Technology in Biomedicine, IEEE Transactions on* 14.2 (2010), pp. 274–283 (cit. on p. 45).

[222] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. “epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition”. In: *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on.* IEEE. 2009, pp. 1–9 (cit. on p. 45).

[223] Ling Bao and Stephen S Intille. “Activity recognition from user-annotated acceleration data”. In: *Pervasive computing.* Springer, 2004, pp. 1–17 (cit. on p. 45).

[224] Maja Stikic and Bernt Schiele. “Activity recognition from sparsely labeled data using multi-instance learning”. In: *Location and Context Awareness* (2009), pp. 156–173 (cit. on pp. 44–46).

[225] David Minnen, Thad Starner, Irfan Essa, and Charles Isbell. “Discovering characteristic actions from on-body sensor data”. In: *Wearable computers, 2006 10th IEEE international symposium on.* IEEE. 2006, pp. 11–18 (cit. on p. 45).

[226] Tinghuai Ma, Yong-Deak Kim, Qiang Ma, Meili Tang, and Weican Zhou. "Context-aware implementation based on CBR for smart home". In: *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on.* Vol. 4. IEEE. 2005, pp. 112–115 (cit. on p. 45).

[227] Matthew Brand, Nuria Oliver, and Alex Pentland. "Coupled hidden Markov models for complex action recognition". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on.* IEEE. 1997, pp. 994–999 (cit. on p. 45).

[228] Tim LM van Kasteren, Gwenn Englebienne, and Ben JA Kröse. "Hierarchical activity recognition using automatically clustered actions". In: *Ambient Intelligence.* Springer, 2011, pp. 82–91 (cit. on p. 45).

[229] Xiao-Qin Cao and Zhi-Qiang Liu. "Human motion detection using Markov random fields". In: *Journal of Ambient Intelligence and Humanized Computing* 1.3 (2010), pp. 211–220 (cit. on p. 45).

[230] Christian Wojek, Kai Nickel, and Rainer Stiefelhagen. "Activity recognition and room-level tracking in an office environment". In: *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on.* IEEE. 2006, pp. 25–30 (cit. on p. 45).

[231] Md Kamrul Hasan, Husne Ara Rubaiyeat, Yong-Koo Lee, and Sungyoung Lee. "A reconfigurable HMM for activity recognition". In: *ICACT.* Vol. 8. 2008, pp. 843–846 (cit. on p. 45).

[232] Brian Clarkson and Alex Pentland. "Unsupervised clustering of ambulatory audio and video". In: *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on.* Vol. 6. IEEE. 1999, pp. 3037–3040 (cit. on p. 45).

[233] Thi V Duong, Hung H Bui, Dinh Q Phung, and Svetha Venkatesh. "Activity recognition and abnormality detection with the switching hidden semi-markov model". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on.* Vol. 1. IEEE. 2005, pp. 838–845 (cit. on p. 45).

[234] G Michael Youngblood and Diane J Cook. "Data mining for hierarchical model creation". In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 37.4 (2007), pp. 561–572 (cit. on pp. 45, 46).

[235] David Minnen, Thad Starner, Jamie A Ward, Paul Lukowicz, and Gerhard Tröster. "Recognizing and discovering human actions from on-body sensor data". In: *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on.* IEEE. 2005, pp. 1545–1548 (cit. on p. 45).

[236] Liang Wang, Tao Gu, Xianping Tao, and Jian Lu. *Sensor-based human activity recognition in a multi-user scenario.* Springer, 2009 (cit. on pp. 45, 49, 132).

[237] Pau-Choo Chung and Chin-De Liu. "A daily behavior enabled hidden Markov model for human behavior understanding". In: *Pattern Recognition* 41.5 (2008), pp. 1572–1580 (cit. on pp. 45, 49, 144).

[238] Nam Nguyen, Svetha Venkatesh, and Hung Bui. "Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association". In: *BMVC 2006: Proceedings of the 17th British Machine Vision Conference*. British Machine Vision Association. 2006, pp. 1239–1248 (cit. on p. 45).

[239] Shai Fine, Yoram Singer, and Naftali Tishby. "The hierarchical hidden Markov model: Analysis and applications". In: *Machine learning* 32.1 (1998), pp. 41–62 (cit. on p. 45).

[240] Nam T Nguyen, Dinh Q Phung, Svetha Venkatesh, and Hung Bui. "Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 2. IEEE. 2005, pp. 955–960 (cit. on p. 45).

[241] Kuo-Chung Hsu, Yi-Ting Chiang, Gu-Yang Lin, Ching-Hu Lu, Jane Yung-Jen Hsu, and Li-Chen Fu. "Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home". In: *Trends in Applied Intelligent Systems*. Springer, 2010, pp. 417–426 (cit. on p. 45).

[242] Maryam Mahdaviani and Tanzeem Choudhury. "Fast and scalable training of semi-supervised crfs with application to activity recognition". In: *Advances in Neural Information Processing Systems*. 2008, pp. 977–984 (cit. on p. 45).

[243] Douglas L Vail, Manuela M Veloso, and John D Lafferty. "Conditional random fields for activity recognition". In: *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*. ACM. 2007, p. 235 (cit. on p. 45).

[244] Liyue Zhao, Xi Wang, Gita Sukthankar, and Rahul Sukthankar. "Motif discovery and feature selection for crf-based activity recognition". In: *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE. 2010, pp. 3826–3829 (cit. on p. 45).

[245] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. "Accurate activity recognition in a home setting". In: *Proceedings of the 10th international conference on Ubiquitous computing*. ACM. 2008, pp. 1–9 (cit. on p. 45).

[246] Thomas G Dietterich, Adam Ashenfelter, and Yaroslav Bulatov. "Training conditional random fields via gradient tree boosting". In: *Proceedings of the twenty-first international conference on Machine learning*. ACM. 2004, p. 28 (cit. on p. 45).

[247] Douglas L Vail and Manuela M Veloso. "Feature Selection for Activity Recognition in Multi-Robot Domains." In: *AAAI*. Vol. 8. 2008, pp. 1415–1420 (cit. on p. 45).

[248] Lin Liao, Dieter Fox, and Henry Kautz. "Extracting places and activities from gps traces using hierarchical conditional random fields". In: *The International Journal of Robotics Research* 26.1 (2007), pp. 119–134 (cit. on p. 45).

[249] Yuri Ivanov, Aaron F Bobick, et al. "Recognition of visual activities and interactions by stochastic parsing". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 22.8 (2000), pp. 852–872 (cit. on p. 45).

[250] Darnell Moore and Irfan Essa. "Recognizing multitasked activities from video using stochastic context-free grammar". In: *AAAI/IAAI*. 2002, pp. 770–776 (cit. on p. 45).

[251] Sebastian Lühr, Geoff West, and Svetha Venkatesh. "Recognition of emergent human behaviour in a smart home: A data mining approach". In: *Pervasive and Mobile Computing* 3.2 (2007), pp. 95–116 (cit. on p. 45).

[252] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. "Automated handwashing assistance for persons with dementia using video and a partially observable markov decision process". In: *Computer Vision and Image Understanding* 114.5 (2010), pp. 503–519 (cit. on p. 45).

[253] Michael C Mozer. "The neural network house: An environment hat adapts to its inhabitants". In: *Proc. AAAI Spring Symp. Intelligent Environments*. 1998, pp. 110–114 (cit. on p. 45).

[254] Mike Perkowitz, Matthai Philipose, Kenneth Fishkin, and Donald J Patterson. "Mining models of human activities from the web". In: *Proceedings of the 13th international conference on World Wide Web*. ACM. 2004, pp. 573–582 (cit. on p. 45).

[255] Paulito Palmes, Hung Keng Pung, Tao Gu, Wenwei Xue, and Shaxun Chen. "Object relevance weight pattern mining for activity recognition and segmentation". In: *Pervasive and Mobile Computing* 6.1 (2010), pp. 43–57 (cit. on pp. 45, 46).

[256] Parisa Rashidi and Diane J Cook. "Mining sensor streams for discovering human activity patterns over time". In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE. 2010, pp. 431–440 (cit. on pp. 45, 46, 48).

[257] Donghai Guan, Weiwei Yuan, Young-Koo Lee, Andrey Gavrilov, and Sungyoung Lee. "Activity recognition based on semi-supervised learning". In: *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*. IEEE. 2007, pp. 469–475 (cit. on p. 45).

[258] Parisa Rashidi and Diane J Cook. "Ask me better questions: active learning queries based on rule induction". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2011, pp. 904–912 (cit. on p. 45).

[259] Tâm Huynh and Bernt Schiele. "Unsupervised discovery of structure in activity data using multiple eigenspaces". In: *Location-and Context-Awareness*. Springer, 2006, pp. 151–167 (cit. on p. 45).

[260] Tracy S Barger, Donald E Brown, and Majd Alwan. "Health-status monitoring through analysis of behavioral patterns". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 35.1 (2005), pp. 22–27 (cit. on p. 45).

[261] Jian Pei, Jiawei Han, and Wei Wang. "Constraint-based sequential pattern mining: the pattern-growth methods". In: *Journal of Intelligent Information Systems* 28.2 (2007), pp. 133–160 (cit. on pp. 45, 160).

[262] Edwin O Heierman III and Diane J Cook. "Improving home automation by discovering regularly occurring device usage patterns". In: *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE. 2003, pp. 537–540 (cit. on p. 45).

[263] Raffay Hamid, Siddhartha Maddi, Amos Johnson, Aaron Bobick, Irfan Essa, and Charles Isbell. "A novel sequence representation for unsupervised analysis of human activities". In: *Artificial Intelligence* 173.14 (2009), pp. 1221–1244 (cit. on p. 45).

[264] Tia Gao, Tammara Massey, Leo Selavo, David Crawford, Bor-rong Chen, Konrad Lorincz, Victor Shnayder, Logan Hauenstein, Foad Dabiri, James Jeng, et al. "The advanced health and disaster aid network: A light-weight wireless medical system for triage". In: *Biomedical Circuits and Systems, IEEE Transactions on* 1.3 (2007), pp. 203–216 (cit. on p. 45).

[265] Dimitrios Lymberopoulos, Athanasios Bamis, and Andreas Savvides. "Extracting spatiotemporal human activity patterns in assisted living using a home sensor network". In: *Universal Access in the Information Society* 10.2 (2011), pp. 125–138 (cit. on p. 45).

[266] Cristian Sminchisescu, Atul Kanaujia, and Dimitris Metaxas. "Conditional models for contextual human motion recognition". In: *Computer Vision and Image Understanding* 104.2 (2006), pp. 210–220 (cit. on p. 46).

[267] Tao Gu, Shaxun Chen, Xianping Tao, and Jian Lu. "An unsupervised approach to activity recognition and segmentation based on object-use fingerprints". In: *Data & Knowledge Engineering* 69.6 (2010), pp. 533–544 (cit. on p. 46).

[268] Michael Mozer. "Lessons from an adaptive house". PhD thesis. Architectural Engineering, 2004 (cit. on p. 46).

[269] Hani Hagras, Victor Callaghan, Martin Colley, Graham Clarke, Anthony Pounds-Cornish, and Hakan Duman. "Creating an ambient-intelligence environment using embedded agents". In: *Intelligent Systems, IEEE* 19.6 (2004), pp. 12–20 (cit. on p. 46).

[270] Raghavendra V Kulkarni, Anna Förster, and Ganesh Kumar Venayagamoorthy. "Computational intelligence in wireless sensor networks: A survey". In: *Communications Surveys & Tutorials, IEEE* 13.1 (2011), pp. 68–96 (cit. on p. 46).

[271] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM computing surveys (CSUR)* 41.3 (2009), p. 15 (cit. on p. 47).

[272] Russ Dewey. *Defining Abnormal Behavior*. 2007. URL: http://www.intropsych.com/ch12_abnormal/defining_abnormal_behavior.html (visited on 07/03/2015) (cit. on p. 47).

[273] An C Tran, Stephen Marsland, Jens Dietrich, Hans W Guesgen, and Paul Lyons. "Use cases for abnormal behaviour detection in smart homes". In: *Aging Friendly Technology for Health and Independence*. Springer, 2010, pp. 144–151 (cit. on p. 47).

[274] Jie Yin, Qiang Yang, and Jeffrey Junfeng Pan. “Sensor-based abnormal human-activity detection”. In: *Knowledge and Data Engineering, IEEE Transactions on* 20.8 (2008), pp. 1082–1090 (cit. on p. 48).

[275] Vikramaditya Jakkula, Diane J Cook, et al. “Anomaly detection using temporal data mining in a smart home environment”. In: *Methods of information in medicine* 47.1 (2008), pp. 70–75 (cit. on p. 48).

[276] Asier Aztiria, Juan Carlos Augusto, Rosa Basagoiti, and Alberto Izaguirre. “Accurate temporal relationships in sequences of user behaviours in intelligent environments”. In: *Ambient Intelligence and Future Trends-International Symposium on Ambient Intelligence (ISAmI 2010)*. Springer. 2010, pp. 19–27 (cit. on p. 48).

[277] GE Wu and Shuwan Xue. “Portable preimpact fall detector with inertial sensors”. In: *Neural Systems and Rehabilitation Engineering, IEEE Transactions on* 16.2 (2008), pp. 178–183 (cit. on p. 48).

[278] Chin-Feng Lai, Sung-Yen Chang, Han-Chieh Chao, and Yueh-Min Huang. “Detection of cognitive injured body region using multiple triaxial accelerometers for elderly falling”. In: *Sensors Journal, IEEE* 11.3 (2011), pp. 763–770 (cit. on p. 48).

[279] Xiaodan Zhuang, Jing Huang, Gerasimos Potamianos, and Mark Hasegawa-Johnson. “Acoustic fall detection using Gaussian mixture models and GMM supervectors”. In: *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE. 2009, pp. 69–72 (cit. on p. 48).

[280] Majd Alwan, Prabhu Jude Rajendran, Steve Kell, David Mack, Siddharth Dalal, Matt Wolfe, and Robin Felder. “A smart and passive floor-vibration based fall detector for elderly”. In: *Information and Communication Technologies, 2006. ICTTA’06. 2nd*. Vol. 1. IEEE. 2006, pp. 1003–1007 (cit. on p. 48).

[281] Homa Foroughi, Aabed Naseri, Alireza Saberi, and Hadi Sadoghi Yazdi. “An eigenspace-based approach for human fall detection using integrated time motion image and neural network”. In: *Signal Processing, 2008. ICSP 2008. 9th International Conference on*. IEEE. 2008, pp. 1499–1503 (cit. on p. 48).

[282] Guangyi Shi, Cheung Shing Chan, Wen Jung Li, Kwok-Sui Leung, Yuexian Zou, and Yufeng Jin. “Mobile human airbag system for fall protection using MEMS sensors and embedded SVM classifier”. In: *Sensors Journal, IEEE* 9.5 (2009), pp. 495–503 (cit. on p. 48).

[283] Sebastian Fudickar and Bettina Schnor. “KopAL–A Mobile Orientation System for Dementia Patients”. In: *Intelligent Interactive Assistance and Mobile Multimedia Computing*. Springer, 2009, pp. 109–118 (cit. on p. 48).

[284] Jie Wan, Caroline Byrne, Gregory MP O’Hare, and Michael J O’Grady. “Orange alerts: Lessons from an outdoor case study”. In: *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on*. IEEE. 2011, pp. 446–451 (cit. on p. 48).

[285] Chung-Chih Lin, Ming-Jang Chiu, Chun-Chieh Hsiao, Ren-Guey Lee, and Yuh-Show Tsai. “Wireless health care service system for elderly with dementia”. In: *Information Technology in Biomedicine, IEEE Transactions on* 10.4 (2006), pp. 696–704 (cit. on p. 48).

[286] Erik Crombag, Ir Rinus Plasmeijer, and Peter Lucas. “Monitoring the elderly using real time location sensing”. PhD thesis. Master’s thesis, RADBOUD UNIVERSITY, 2009 (cit. on p. 48).

[287] TLM Van Kasteren, Gwenn Englebienne, and Ben JA Kröse. “Transferring knowledge of activity recognition across sensor networks”. In: *Pervasive computing.* Springer, 2010, pp. 283–300 (cit. on p. 48).

[288] Parisa Rashidi and Diane J Cook. “Activity knowledge transfer in smart environments”. In: *Pervasive and Mobile Computing* 7.3 (2011), pp. 331–343 (cit. on p. 48).

[289] Parisa Rashidi and Diane J Cook. “Domain selection and adaptation in smart homes”. In: *Toward Useful Services for Elderly and People with Disabilities.* Springer, 2011, pp. 17–24 (cit. on p. 48).

[290] Jose Antonio Iglesias, Plamen Angelov, Agapito Ledezma, and Araceli Sanchis. “Creating evolving user behavior profiles automatically”. In: *Knowledge and Data Engineering, IEEE Transactions on* 24.5 (2012), pp. 854–867 (cit. on p. 48).

[291] Joseph Modayil, Tongxin Bai, and Henry Kautz. “Improving the recognition of interleaved activities”. In: *Proceedings of the 10th international conference on Ubiquitous computing.* ACM. 2008, pp. 40–43 (cit. on p. 49).

[292] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. “Recognizing independent and joint activities among multiple residents in smart environments”. In: *Journal of ambient intelligence and humanized computing* 1.1 (2010), pp. 57–63 (cit. on p. 49).

[293] Aaron S Crandall and Diane J Cook. “Coping with multiple residents in a smart environment”. In: *Journal of Ambient Intelligence and Smart Environments* 1.4 (2009), pp. 323–334 (cit. on p. 49).

[294] Clifton Phua, Kelvin Sim, and Jit Biswas. “Multiple People Activity Recognition using Simple Sensors.” In: *PECCS.* 2011, pp. 224–231 (cit. on p. 49).

[295] Romain Endelin, Stéphane Renouard, Thibaut Tiberghien, Hamdi Aloulou, and Mounir Mokhtari. “Behavior recognition for elderly people in large-scale deployment”. In: *Inclusive Society: Health and Wellbeing in the Community, and Care at Home.* Springer, 2013, pp. 61–68 (cit. on pp. 49, 133).

[296] Datong Chen, Jie Yang, and Howard D Wactlar. “Towards automatic analysis of social interaction patterns in a nursing home environment from video”. In: *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval.* ACM. 2004, pp. 283–290 (cit. on p. 49).

[297] Paul Panek, Walter Hlauschek, Manfred Schrenk, Katharina Werner, and Wolfgang L Zagler. "Experiences from user centric engineering of ambient assisted living technologies in the living lab schwechat". In: *Concurrent Enterprising (ICE), 2011 17th International Conference on*. IEEE. 2011, pp. 1–8 (cit. on p. 49).

[298] Diane J Cook. "Learning setting-generalized activity models for smart spaces". In: *IEEE intelligent systems* 2010.99 (2010), p. 1 (cit. on pp. 49, 144).

[299] CD Nugent, MD Mulvenna, Xin Hong, and S Devlin. "Experiences in the development of a smart lab". In: *International Journal of Biomedical Engineering and Technology* 2.4 (2009), pp. 319–331 (cit. on p. 49).

[300] Mukhtiar Memon, Stefan Rahr Wagner, Christian Fischer Pedersen, Femina Hassan Aysha Beevi, and Finn Overgaard Hansen. "Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes". In: *Sensors* 14.3 (2014), pp. 4312–4341 (cit. on p. 49).

[301] James McNaull, Juan Carlos Augusto, Maurice Mulvenna, and Paul McCullagh. "Data and information quality issues in ambient assisted living systems". In: *Journal of Data and Information Quality (JDIQ)* 4.1 (2012), p. 4 (cit. on p. 50).

[302] Karen Henricksen and Jadwiga Indulska. "Modelling and using imperfect context information". In: *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*. IEEE. 2004, pp. 33–37 (cit. on p. 50).

[303] Sidney Katz, Amasa B Ford, Roland W Moskowitz, Beverly A Jackson, and Marjorie W Jaffe. "Studies of illness in the aged: the index of ADL: a standardized measure of biological and psychosocial function". In: *Jama* 185.12 (1963), pp. 914–919 (cit. on p. 50).

[304] M Powell Lawton. "Scales to measure competence in everyday activities." In: *Psychopharmacology bulletin* 24.4 (1987), pp. 609–614 (cit. on p. 50).

[305] Rosalie A Kane, Robert L Kane, et al. *Assessing the elderly*. LexingtonBooks, 1981 (cit. on p. 50).

[306] Gerda G Fillenbaum. *Multidimensional functional assessment of older adults: The Duke Older Americans Resources and Services procedures*. Psychology Press, 2013 (cit. on p. 50).

[307] Thomas Kleinberger, Andreas Jedlitschka, Holger Storf, Silke Steinbach-Nordmann, and Stephan Prueckner. "An approach to and evaluations of assisted living systems using ambient intelligence for emergency monitoring and prevention". In: *Universal Access in Human-Computer Interaction. Intelligent and Ubiquitous Interaction Environments*. Springer, 2009, pp. 199–208 (cit. on p. 50).

[308] Victor R Basili. "Software modeling and measurement: the Goal/Question/Metric paradigm". In: (1992) (cit. on p. 50).

[309] Bessam Abdulrazak and Yasir Malik. "Review of challenges, requirements, and approaches of pervasive computing system evaluation". In: *IETE Technical Review* 29.6 (2012), pp. 506–522 (cit. on p. 51).

[310] Roy Thomas Fielding. "Architectural styles and the design of network-based software architectures". PhD thesis. University of California, Irvine, 2000 (cit. on p. 94).

[311] James Martin and Savant Institute. *Managing the data-base environment*. Prentice-Hall Englewood Cliffs (NJ), 1983 (cit. on p. 94).

[312] Julia. *Julia Benchmarks*. URL: `http://julialang.org/benchmarks/` (visited on 12/20/2015) (cit. on p. 112).

[313] Sumi Helal, Jae Woong Lee, Shantonu Hossain, Eunju Kim, Hani Hagras, and Diane Cook. "Persim-Simulator for human activities in pervasive spaces". In: *Intelligent Environments (IE), 2011 7th International Conference on*. IEEE. 2011, pp. 192–199 (cit. on pp. 122, 127).

[314] M Javad Akhlaghinia, Ahmad Lotfi, Caroline Langensiepen, and Nasser Sherkat. "Single-occupancy simulator for ambient intelligent environment". In: *Journal of Internet Technology* 9.4 (2008), pp. 333–338 (cit. on p. 122).

[315] Stewart Robinson. "Simulation: The Practice of Model Development and Use". In: (2004) (cit. on p. 122).

[316] Fabio Veronese, Sara Comai, Matteo Matteucci, and Fabio Salice. "Method, design and implementation of a multiuser indoor localization system with concurrent fault detection". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering). 2014, pp. 100–109 (cit. on p. 132).

[317] Kaikai Liu, Xinxin Liu, and Xiaolin Li. "Guoguo: Enabling fine-grained indoor localization via smartphone". In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM. 2013, pp. 235–248 (cit. on p. 132).

[318] Loic Sevrin, Norbert Noury, Nacer Abouchi, Fabrice Jumel, Bertrand Massot, and Jacques Saraydaryan. "Characterization of a multi-user indoor positioning system based on low cost depth vision (Kinect) for monitoring human activity in a smart home". In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE. 2015, pp. 5003–5007 (cit. on p. 132).

[319] Juan Ye and Graeme Stevenson. "Semantics-driven multi-user concurrent activity recognition". In: *Ambient Intelligence*. Springer, 2013, pp. 204–219 (cit. on p. 132).

[320] Diane J Cook, Aaron Crandall, Geetika Singla, and Brian Thomas. "Detection of social interaction in smart spaces". In: *Cybernetics and Systems: An International Journal* 41.2 (2010), pp. 90–104 (cit. on p. 132).

[321] Ahmed Nait Aicha, Gwenn Englebienne, and Ben Kröse. “How Lonely is Your Grandma?: Detecting the Visits to Assisted Living Elderly from Wireless Sensor Network Data”. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. UbiComp ’13 Adjunct. ACM, 2013, pp. 1285–1294 (cit. on p. 132).

[322] **Romain Endelin**, Mounir Mokhtari, Bessam Abdulrazak, Hamdi Aloulou, Firas Kaddachi, and Joaquim Bellmunt. “Activity Recognition Enhancement Based on Groundtruth: Introducing Accuracy and Precision Metrics”. In: *Thirteenth International Conference on Ubiquitous Intelligence and Computing (UIC 2016)*. IEEE, submitted, 2016 (cit. on p. 143).

[323] Rachid Kadouche, Bessam Abdulrazak, Mounir Mokhtari, Sylvain Giroux, and Hélène Pigot. “Semantic Matching Framework for handicap situation detection in smart environments”. In: *Journal of Ambient Intelligence and Smart Environments* 1.3 (2009), pp. 223–234 (cit. on p. 144).

[324] Eliyahu M Goldratt, Jeff Cox, and David Whitford. *The goal: a process of ongoing improvement*. Vol. 3. North River Press Great Barringtonˆ eMA MA, 2004 (cit. on p. 158).

List of Figures

1.1 Population pyramids around the world 4
1.2 Persons living with elderly people 5
1.3 The relation between elderly people and caregivers 5
1.4 The concepts of the Open Interconnect Consortium Organisation 12

2.1 Ambient Intelligence Services 28
2.2 The 3-part architecture of a pervasive system 33
2.3 The 3-part architecture of a pervasive system: centralized or distributed . . 33
2.4 Sensor networks: wired and wireless 34
2.5 The DIKW pyramid 39
2.6 Qualitative Temporal Logic with Allen's Interval Logic 41
2.7 Qualitative Spatial Logic with RCC8 41
2.8 The matrix of knowledge extraction methods. 44
2.9 Use cases for abnormal human behaviours 47

3.1 Elders studied in the questionnaire 54
3.2 Daily environment of the elders 55
3.3 Situation of the caregivers 56
3.4 The roles of caregivers 57
3.5 Distance between the caregivers and the elders 57
3.6 Main activities of elderly people 58
3.7 Technical aids used by the elders 59
3.8 Limitations of activities 60
3.9 Difficulties in taking medications 61
3.10 Difficulties in dressing 62
3.11 Difficulties in eating 62
3.12 Difficulties in cleaning the house 63
3.13 Difficulties in performing administrative activities 64
3.14 Products of daily life used by the elders 64
3.15 Outdoor activities performed by the elders 65
3.16 Transportations used by the elders 66
3.17 Difficulties in using phone 67
3.18 Feeling of security inside elderly people' home 68
3.19 Feeling of security in the outside 69
3.20 Monitoring services used by the elders 70
3.21 Home-care services 71
3.22 Elderly people's opinion about home services 72
3.23 Financial helps received by elderly people 73
3.24 Services requested by the elders 74

4.1 Design of the UbiSMART framework . . . 80
4.2 UbismartV2 . . . 82
4.3 A Raspberry Pi — The UbiSMART gateways . . . 88
4.4 Simple sensors using X10-RF technology . . . 89
4.5 A prototype of bed sensor . . . 89
4.6 A USB 3G modem . . . 90
4.7 The concept of reverse proxying . . . 92
4.8 UbiSMART Login Interface . . . 94
4.9 UbiSMART House Menu . . . 95
4.10 Visualizing Activities in UbiSMART . . . 96
4.11 Visualizing a Nursing Home Corridor in UbiSMART . . . 96
4.12 Visualizing House in UbiSMART . . . 97
4.13 Visualizing Sensor Events in UbiSMART . . . 98
4.14 Statistical visualisations in UbiSMART . . . 98
4.15 UbiSMART Alert on Smartphone . . . 99
4.16 UbiSMART Alert on SmartTV . . . 100
4.17 Our gateway monitoring interface . . . 102
4.18 Load test of UbiSMART V3 . . . 103
4.19 Describing a house through a web interface . . . 104
4.20 The different modules in place in UbiSMART V3 . . . 105

5.1 Jupyter Notebooks . . . 113
5.2 File architecture of Knot . . . 113
5.3 Illustrating the plot method . . . 118
5.4 House to be simulated . . . 125

6.1 A Multiuser Situation . . . 131
6.2 Automatic detection of multiuser situations . . . 132

7.1 Sensor events over a day in a house . . . 145
7.2 Sensor events — Resident taking lunch . . . 146
7.3 Incorrect inference: Reasoner trying to be too Precise . . . 146
7.4 Incorrect inference: Faulty rule . . . 147
7.5 Incorrect inference: Reasoner not being Precise enough . . . 147
7.6 Incorrect inference: Reasoner facing confusing events . . . 148
7.7 Relation between Real Activities, Inferred Activities, and Sensor Events . . . 150
7.8 Smartphone application to gather ground-truth . . . 151
7.9 Illustrating the relation between Accuracy and Precision . . . 154
7.10 Hierarchical activities in the UbiSMART reasoner . . . 155

8.1 Defining sensor models with the ORM . . . 163

9.1 Topology of the first house deployed . . . 174
9.2 Overview of ground-truth for one resident . . . 176
9.3 Plotting events and inferred activities . . . 182
9.4 Multiuser Situations Over Two Days . . . 183

9.5 Density of Multiuser Situations During the Day 184

List of Tables

2.1 Applications of Ambient Intelligence for Healthcare 29
2.2 Existing research platforms in Ambient Assisted Living 31
2.3 Ambient Sensors Used in Smart Environments 35
2.4 A comparative table of semantic reasoners (2012) 43
2.5 Supervised vs. Unsupervised and Static- vs. Sequential-Based Learning . . 45

4.1 Technical architecture of UbiSMART V2 83

5.1 Comparing programming languages 112
5.2 Dataframes 116
5.3 Simulated events 126

6.1 Original Dataset of events 134
6.2 Events grouped by sensors 135
6.3 Events resampled to uniform frequency, per room 136
6.4 Presence of the resident in rooms 138
6.5 Multiuser situations 139
6.6 Location of the resident 140
6.7 Final dataset — location of the resident 140

7.1 Illustrating Accurate/Inaccurate inferences 149

8.1 Existing components of the house factory 162
8.2 Existing condition patterns in the rules factory 165
8.3 Requirements for each rule template 166

9.1 Amount of data obtained in our deployment 174
9.2 Amount of data obtained in the deployment from Handco 175
9.3 Summary of ground-truth observations from our deployments 175
9.4 Definition of the scenario to be executed 177
9.5 Events generated by the simulator 179
9.6 Activities inferred from the simulated events 180
9.7 Events from dataset 182
9.8 Topology of the experiment 185
9.9 Scores of the reasoner 187
9.10 Results with $(A = 3, P = 1)$ and $(A = 1, P = 1)$, *before* and after the calibration 187
9.11 Definition of the house 188
9.12 Definition of the activities 191
9.13 Detecting missing sensors — Defining a house 192
9.14 Detecting missing sensors — Defining rules 193

9.15 Detecting missing sensors — Results 193
9.16 Examples of conflicting rules detection 194

List of Listings

5.1 Notation3 (N3) rule, to be inferred by the Euler Rule Engine 111
5.2 Loading Knot 114
5.3 Exporting from Database to CSV format 115
5.4 The Plot function 117
5.5 An example coroutine 119
5.6 The reasoner coroutine 120
5.7 The Dataframe source 120
5.8 The Dataframe sink 121
5.9 Simulating a motion sensor 124
5.10 Example scenario for the simulator 125
5.11 Executing the scenario with the simulator 125

8.1 Defining a house using the Houses Factory 164
8.2 Defining a rule using the Rules Factory 165

9.1 Defining a scenario, and running the simulator 178
9.2 Running the reasoner 179
9.3 Input events file 180
9.4 Preprocessing events 181
9.5 Plotting events and inferred activities 181
9.6 Defining the house, using House Factory 189
9.7 Generated ontology of the house 190
9.8 Defining activities, using Rule Factory 191
9.9 Generated ontology of the activities 192

B.1 Grouping events by sensor 213
B.2 Resampling events to uniform frequency, per sensor 214
B.3 Inferring presence in rooms 215
B.4 Infering multiuser situations 215
B.5 Concluding on the resident's location 216
B.6 Reducing the location list 216

Glossary

Activity Recognition The process of recognizing the activities being performed by the end-user. i, 17, 18, 20, 21, 25, 36, 37, 40, 50, 51, 110, 127, 132, 133, 141, 143, 144, 157, 181, 183, 184, 193, 194, 197–200

context-awareness The ability for a computer to maintain its knowledge of a certain context in his environment. 14, 15, 18, 19, 25, 36

ground-truth Direct observations of the end-users activities in a real deployment. ii, 20, 110, 175, 176, 184–186, 246, 249

Knot A framework for prototyping activity recognition, developed during this thesis work. 109–112, 114–117, 119–122, 127, 128, 161, 163, 169, 173, 177, 180–182, 184, 193, 194, 199, 213

pervasive computing Paradigm where the computer's presence is ambient, and adapts to the end-user. 9–11, 13, 14, 16, 17, 80

RDFLib Python library for managing RDF content dynamically. 112, 127, 169

smart home A residence equipped with technology, and using a certain level of intelligence to adapt to the resident's needs. 15–17, 20, 25–27, 31, 51, 66, 84, 86, 87, 92, 108, 110, 193

the Euler Rule Engine Ontological Reasoning Engine, used in UbiSMART. 19, 111, 121, 127, 156, 251

Ubigate Software facilitating the gateway's communications communication with the UbiSMART server. 91, 101, 106, 107, 199, 200

ubiquitous computing Synonym, pervasive computing. 10, 11

UbiSMART A framework for Ambient Assisted Living. 19, 199, 256

UbiSMART V2 The previous version of UbiSMART. 81, 83, 84, 86, 93, 95, 102, 103, 249

UbiSMART V3 The current version of UbiSMART, partly developed during this thesis work. 81, 84–87, 93, 95, 102, 103, 105, 108, 246

Acronyms

AAL Ambient Assisted Living. i, 9, 13–21, 25, 28, 30–32, 35–37, 39–41, 43, 44, 46, 48–51, 54, 58, 59, 66, 67, 73, 75, 79–81, 86, 87, 105, 106, 108, 109, 122, 128, 132, 143, 144, 157, 161, 162, 164, 166, 169, 193, 197, 198, 200, 201

ADL Activities of daily Living. 6, 14, 39, 46, 50

AI Artificial Intelligence. 40, 41

AmI Ambient Intelligence. 13, 14, 16, 25, 28, 30, 35–37, 108

API Application Programming Interface. 162

BN Bayesian Network. 44

CLI Command-Line Interface. 10

CRF Conditional Random Field. 46

CRUD Create, Read, Update, Delete. 94, 163

CSV Comma-Separated Value. 115, 116, 180

D-HTN Distributed Hierarchical Task Network. 37

DBN Dynamic Bayesian Network. 46, 51

DIKW Data-Information-Knowledge-Wisdom. 39

DL Description Logic. 42, 43

DPWS Devices Profile for Web Services. 36

DST Dempster-Shafer Theory. 43, 44

GQM Goal-Question-Metric. 50

GUI Graphical User Interface. 169

HC-HMM Hierarchical Context Hidden Markov Model. 144

HMI Human-Machine Interactions. 10, 16

HMM Hidden Markov Models. 46, 50, 51, 132

Insee the French National Institute of Statistics and Economic Studies. 3

IoT Internet of Things. i, 9, 11–13, 25, 27, 44, 46, 108

IP Internet Protocol. 11

JSON JavaScript Object Notation. 163, 164, 169, 188

LOCF Last Observation Carried Forward. 133

MQTT MQ Telemetry Transport. 85, 91, 93, 101

MVC Model-View-Controller. 163

N3 Notation3. 19, 42, 110, 111, 127, 191, 192, 251

ORM Object-Relational Mapping. 128, 162, 163, 169

OSGi Open Service Gateway Initiative. 36, 81

OWL Web Ontology Language. 42

PEAS Performance Environment Actuators Sensors. 36, 37

QoL Quality of Life. 81

RDBMS Relational Database Management System. 115

RDF Resource Description Framework. 42, 109, 110, 112, 127, 163, 164, 169

RF Radio Frequency. 34

RFID Radio Frequency Identification. 49, 166

SOA Service-Oriented Architecture. 36, 81

SVM Support Vector Machine. 46, 144

UbiSMART Ubiquitous Service MAnagement & Reasoning archiTecture. 19, 32, 50, 75, 79–81, 86–88, 93–102, 106–111, 115–117, 121, 127, 145, 155, 156, 173, 176, 184, 185, 193, 194, 199, 246, *Glossary:* UbiSMART

UX User eXperience. 107

WIMP Window, Icon, Menu, Pointer. 10

WMN Wireless Mesh Network. 35

WSN Wireless Sensor Network. 35

Doctoral thesis to be submitted in March, 29th 2016, and defended in June, 2nd 2016.