Framework for ambient assistive living : handling dynamism and uncertainty in real time semantic services provisioning

Hamdi Aloulou

THÈSE DE DOCTORAT CONJOINT
TELECOM SUDPARIS / UNIVERSITÉ PIERRE & MARIE CURIE

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique de Paris

Présentée par

Hamdi ALOULOU

Pour obtenir le grade de

DOCTEUR
de
TELECOM SUDPARIS

Sujet de la thèse :

Environnement Logiciel pour l'Assistance à l'Autonomie à Domicile : Gestion de la Dynamique et de l'Incertitude pour la Fourniture Sémantique en Temps Réel de Services d'Assistance

Soutenue le 25 Juin 2013

devant le jury composé de :

M. Jacques Demongeot	Rapporteur	Prof, Université Joseph Fourier, France
M. Nick Hine	Rapporteur	Prof, Université de Dundee, Royaume-Uni
Mme. Amal El Fallah Seghrouchni	Examinateur	Prof, Université Pierre & Marie Curie, France
M. Bessam Abdulrazak	Examinateur	Prof, Université de Sherbrooke, Canada
M. Philippe Fraisse	Examinateur	Prof, Université Montpellier 2, France
M. Mounir Mokhtari	Directeur de thèse	Prof, Institut Mines-Télécom, France

Thèse $n°$ 2013TELE0016

DOCTOR OF PHILOSOPHY (PhD) THESIS
TELECOM SUDPARIS / PIERRE ET MARIE CURIE UNIVERSITY

Specialization

COMPUTER SCIENCE

Paris doctoral school of Computing, Communications and Electronic

Presented by

Hamdi ALOULOU

Submitted for the partial requirement of

Doctor of Philosophy
from
TELECOM SUDPARIS

Thesis subject :

Framework for Ambient Assistive Living: Handling Dynamism and Uncertainty in Real Time Semantic Services Provisioning

June 25 2013

Commitee:

Mr. Jacques Demongeot	Reviewer	Prof, Joseph Fourier University, France
Mr. Nick Hine	Reviewer	Prof, Dundee University, United Kingdom
Ms. Amal El Fallah Seghrouchni	Examiner	Prof, University Pierre & Marie Curie, France
Mr. Bessam Abdulrazak	Examiner	Prof, University of Sherbrooke, Canada
Mr. Philippe Fraisse	Examiner	Prof, Montpellier 2 University, France
Mr. Mounir Mokhtari	Thesis director	Prof, Mines Télécom Institute, France

Thèse $n°$ 2013TELE0016

Institut Mines-Telecom
Image & Pervasive Access Lab
CNRS (UMI 2955), France
I^2R / A*STAR, Singapore

1 Fusionopolis Way
#21-01 Connexis (South Tower)
Singapore 138632

To my father Youssef,

To my mother Rafia,

To my wife Sahar,

To my beloved brother and sister,

Thank you for your sacrifices and patience.

Indeed, with hardship will be ease.

Acknowledgments

First of all, I would like to thank all the people who supported and collaborated with me to achieve this work. Without your help I could not be at this stage today.

I wish to extend my sincerest gratitude to my thesis director, Professor Mounir Mokhtari, for giving me the opportunity to realize this work and for his precious advices, ideas and support that brought this work to completion. It was really a wonderful experience to work with him and I would like to thank him for the confidence he had in me and for the freedom and responsibility I received in my work.

I am grateful to Professor Jacques Demongeot and Professor Nick Hine for accepting to review the manuscript and to the rest of my defense jury: Professor Amal El Fallah Seghrouchni, Professor Bessam Abdulrazak and Professor Philippe Fraisse for their interest in my research and for accepting to evaluate it.

A special thank to Professor Daniel Racoceanu, the Image and Pervasive Access Lab (IPAL) director, for welcoming me and for the support I have received. I would like also to thank all my friends in the lab for their help and support all over my stay especially my team-mate Thibaut Tiberghien and my colleagues Antoine Fagette and Stéphane Rigaud.

A part of this research work was realized in close collaboration with the Institute for Infocomm Research (I^2R) and the National University of Singapore (NUS). I would like to thank Doctor Jit Biswas, Professor Jin Song Dong and their teams for the collaboration we had and the work they have achieved. We have also collaborated with Khoo Teck Puat Hospital and PeaceHaven nursing home in Singapore. A special thank to the staff for their collaboration and for welcoming us in their premises. I will not forget all the residents who were involved in our trials. I hope this work will help to improve their well-being and quality of life.

Finally, I would like to thank my family and my friends for their understanding and support during the past three years.

Thanks to all of you.

Framework for Ambient Assistive Living: Handling Dynamism and Uncertainty in Real Time Semantic Services Provisioning

Abstract

The heterogeneity of the environments as well as the diversity of patients' needs and profiles are major constraints that challenge the spread of ambient assistive living (AAL) systems. AAL environments are usually evolving by the introduction or the disappearance of sensors, devices and assistive services to respond to the evolution of patients' conditions and human needs. In addition, each patient has a specific profile that influences the choice of interaction devices and requires particular assistive actions. The selection of the required assistive actions affects the decision on the set of sensors that needs to be installed. Therefore, a generic framework that is able to adapt to such dynamic environments and to integrate new sensors, devices and assistive services at runtime is required. Implementing such a dynamic aspect may produce an uncertainty derived from technical problems related to sensors reliability or network problems. This uncertainty impacts the information and the events received by the framework. It is also related to human behaviors, where the situation of the assisted person is imprecise and the system is not able to classify his activity. Therefore, a notion of uncertain should be introduced in context representation and decision making in order to deal with this problem.

During this thesis, I have developed a dynamic and extendible framework able to adapt to different environments and patients' needs. It allows the integration of new assistive services, sensors and interaction devices at runtime, and then their inclusion into the reasoning process in order to provide the appropriate assistive services for end-users when needed. This was achieved based on my proposed approach of semantic Plug&Play mechanism. I have used different approaches, mainly the modular approach with the use of the Open Service Gateway initiative (OSGi) for implementation, the declarative approach, using the semantic web technologies for environment representation and for reasoning, and the Device Profile for Web Services (DPWS) mechanism for sensors and devices discovery

and interaction with the framework. In order to handle the problem of uncertain information related to technical problems, I have proposed an approach for uncertainty measurement based on intrinsic characteristics of the sensors and their functional behaviors, then I have provided a model of semantic representation and reasoning under uncertainty coupled with the Dempster-Shafer Theory of evidence (DST) for decision making.

The developed framework evolved during on field work and real world deployment in a collaborating nursing home. Through our deployment approach, we have identified the principal requirements we have dealt with in this thesis, and performed a technical performance validation of the framework with an analysis of the collected data.

Keywords

Ambient Assistive Living, Dynamic Framework, Uncertainty Handling, Semantic Plug&Play, Real World Deployment, Modular Approach, Declarative Approach, Open Service Gateway initiative, Device Profile for Web Services, Semantic Web Technologies, Dempster-Shafer Theory

Plate-forme pour l'Assistance à l'Autonomie à Domicile : Gestion du Dynamisme et de l'Incertitude pour la Provision Sémantique en Temps Réel de Services d'Assistance

Résumé

L'hétérogénéité des environnements ainsi que la diversité des profils et des besoins des patients représentent des contraintes majeures qui remettent en question l'utilisation à grande échelle des systèmes d'assistance à l'autonomie à domicile (AAL). En effet, afin de répondre à l'évolution de l'état des patients et de leurs besoins humains, les environnements AAL sont en évolution continue par l'introduction ou la disparition de capteurs, de dispositifs d'interaction et de services d'assistance. En outre, chaque patient a son propre profil qui influence le choix des dispositifs d'interaction et nécessite des actions particulières d'assistance. La sélection des actions d'assistance requises influe la décision sur l'ensemble des capteurs qui doivent être installés. Par conséquent, une plate-forme générique et dynamique capable de s'adapter à différents environnements et d'intégrer de nouveaux capteurs, dispositifs d'interaction et services d'assistance est requise. La mise en œuvre d'un tel aspect dynamique peut produire une situation d'incertitude dérivée des problèmes techniques liés à la fiabilité des capteurs ou à des problèmes de réseau. Cette incertitude affecte les informations et les événements reçus par la plate-forme. L'incertitude est également liée à des comportements humains, où la situation de la personne assistée est imprécise et le système n'est pas capable de classer ses activités. Par conséquent, la notion d'incertitude doit être introduite dans la représentation de contexte et la prise de décision afin de faire face à ce problème.

Au cours de cette thèse, j'ai développé une plate-forme dynamique et extensible capable de s'adapter à différents environnements et aux besoins des patients. Elle permet l'intégration de nouveaux services d'assistance, de capteurs et dispositifs d'interaction en temps réel, puis leur inclusion dans le processus de raisonnement afin de fournir aux patients, en cas de besoin, les services d'assistance appropriés. Ceci a été réalisé sur la base de l'approche *Plug&Play* sémantique que j'ai proposé. J'ai utilisé différentes approches, principalement l'approche modulaire, avec l'utilisation de l'*Open Service Gateway initia-*

tive (OSGi), l'approche déclarative, en utilisant les technologies du web sémantique pour la représentation de l'environnement et pour le raisonnement, et le *Device Profile for Web Services (DPWS)* pour la découverte des capteurs et des dispositifs d'interaction et leurs échanges avec la plate-forme. Afin de traiter le problème d'incertitude de l'information lié à des problèmes techniques, j'ai proposé une approche de mesure d'incertitude en utilisant les caractéristiques intrinsèques des capteurs et leurs comportements fonctionnels. J'ai aussi fourni un modèle de représentation sémantique et de raisonnement avec incertitude associé avec la théorie de Dempster-Shafer (DST) pour la prise de décision.

La plate-forme développée a évolué au cours de notre travail sur le terrain et notre déploiement dans le monde réel au sein d'une maison de retraite. Grâce à notre approche de déploiement, nous avons identifié les principaux besoins que nous avons traités dans cette thèse, nous avons aussi effectué une validation technique de la performance de la plate-forme avec une analyse des données collectées.

Mots-clefs

Assistance à l'Autonomie à Domicile, Plate-forme Dynamique, Gestion d'Incertitude, le *Plug&Play* Sémantique, le Déploiement dans le Monde Réel, l'Approche Modulaire, l'Approche Déclarative, l'*Open Service Gateway initiative*, *Device Profile for Web Services*, les Technologies du Web Sémantique, la théorie de Dempster-Shafer

Author's Publications

Journal Papers

H. Aloulou, M. Mokhtari, T. Tiberghien, J. Biswas, and P. Yap, "Real world deployment of assistive living technologies for cognitively impaired people in singapore: Demonstration guidelines," in *IEEE Journal of Biomedical and Health Informatics (J-BHI)*, IEEE, 2013 [major modifications].

H. Aloulou, M. Mokhtari, T. Tiberghien, J. Biswas, C. Phua, J. H. K. Lin, and P. Yap, "Deployment of assistive living technology in a nursing home environment: methods and lessons learned," *BMC Medical Informatics and Decision Making*, vol. 13, no. 1, p. 42, 2013.

M. Mokhtari, **H. Aloulou**, T. Tiberghien, J. Biswas, D. Racoceanu, and P. Yap, "New trends to support independence in persons with mild dementia – a mini-review," in *International Journal of Experimental, Clinical, Behavioural, Regenerative and Technological Gerontology*, vol. 58, pp. 554–563, Karger Publishers, 2012.

Conference Papers

R. Endelin, S. Renouard, T. Tiberghien, **H. Aloulou**, and M. Mokhtari, "Behavior recognition for elderly people in large-scale deployment," in *Inclusive Society: Health and Well being in Ageing-friendly Community (ICOST)*, Springer, 2013.

H. Aloulou, M. Mokhtari, T. Tiberghien, J. Biswas, and J. H. K. Lin, "A semantic plug&play based framework for ambient assisted living," in *Impact Analysis of solutions for chronic disease Prevention and Management (ICOST)*, vol. 7251, pp. 165–172, Springer, 2012.

T. Tiberghien, M. Mokhtari, **H. Aloulou**, and J. Biswas, "Semantic reasoning in context-aware assistive environments to support ageing with dementia," in *Proceedings of the 11th International Semantic Web Conference (ISWC)*, pp. 212–227, Springer, 2012.

T. Tiberghien, M. Mokhtari, **H. Aloulou**, J. Biswas, J. Zhu, and V. Lee, "Handling user interface plasticity in assistive environment: Ubismart framework," in *Toward Useful*

Services for Elderly and People with Disabilities (ICOST), vol. 6719, pp. 256–260, Springer, 2011.

J. Zhu, V. Y. Lee, J. Biswas, M. Mokhtari, T. Tiberghien, and **H. Aloulou**, "Context-aware reasoning engine with high level knowledge for smart home," in *Proceedings of the 1st International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS)*, pp. 292–297, SciTePress, 2011.

J. Biswas, A. A. Phyo Wai, A. Tolstikov, J. H. K. Lin, M. Jayachandran, S. F. V. Foo, V. Y. Lee, C. Phua, J. Zhu, T. H. Huynh, T. Tiberghien, **H. Aloulou**, and M. Mokhtari, "From context to micro-context-issues and challenges in sensorizing smart spaces for assistive living," in *Proceedings of the 2nd International Conference on Ambient Systems, Networks and Technologies (ANT)*, vol. 5, pp. 288–295, Elsevier, 2011.

C. Phua, J. Biswas, A. Tolstikov, V. Foo, W. Huang, M. Jayachandran, A. Aung, P. Roy, **H. Aloulou**, M. Feki, *et al.*, "Plan recognition based on sensor produced micro-context for eldercare," in *Proceedings of The First International Workshop on Context-Awareness in Smart Environments: Background, Achievements and Challenges (CASEbac 2009)*, (Toyama, Japan), pp. 39–46, 2009.

H. Aloulou, M. A. Feki, C. Phua, and J. Biswas, "Efficient incremental plan recognition method for cognitive assistance," in *Ambient Assistive Health and Wellness Management in the Heart of the City (ICOST)*, (Tours, France), p. 225–228, Springer, 07 2009.

Book Chapters

C. Phua, P. Roy, **H. Aloulou**, J. Biswas, A. Tolstikov, S. F. V. Foo, W. Huang, M. A. Feki, J. Maniyeri, A. Chu, *et al.*, *State-of-the-Art Assistive Technology for People with Dementia*, vol. 1, ch. 16, p. 300. IGI Global, 06/2011 2011.

Reports

H. Aloulou, "Deviant activities recognition: An ambient assistive living application," Master's thesis, National Engineering School of Sfax (Tunisia), IPAL Lab (Singapore), 2010.

Table of Contents

Acronyms

AAL Ambient Assistive Living. 3, 12, 14, 15, 18, 19, 21–23, 27–31, 39–45, 49, 50, 76, 79, 101, 123, 128, 129, 163

ADL Activities of Daily Living. 5, 11, 14, 101, 103, 112

AE Assistive Environment. 12, 14, 16

AmI Ambient Intelligence. 13, 14, 42

APIs Application Programming Interfaces. 43

AT Assistive Technology. 12, 14, 18, 19, 27, 29, 40, 41

DBN Dynamic Bayesian network. 81

DMS Device Management System. 58

DOSGi Distributed Open Service Gateway initiative. 36

DPWS Device Profile for Web Services. 28, 37, 38, 55–59, 75, 94, 128, 137–140

DST Dempser-Shafer Theory. 81–85, 90, 91, 95, 129, 162, 167

GDS Global Deterioration Scale. 10, 104, 122

ICT Information and Communication Technology. 11–13, 17

IE Intelligent Environments. 12, 13

JMS Java Messaging Service. 37

N3 Notation3. 41, 44, 60, 66, 68, 77, 147

OSGi Open Service Gateway initiative. 31–39, 43, 51–54, 57–59, 65, 91, 96, 128, 138–141, 163, 167

PIR Passive Infra-Red. 53, 60, 65, 67, 140

QoC Quality of Context. 73, 76

QoI Quality of Information. 44

QoL Quality of Life. 12

R4 OSGi Specification Release 4. 34–36, 43

RDF Resource Description Framework. 40, 41, 76, 77

RFID Radio-Frequency IDentification. 19, 107

SMS Sensor Management System. 57, 58, 140

SOA Service Oriented Approach. 23, 29–31, 36, 40, 43

SOAP Simple Object Access Protocol. 31

UbiSMART Ubiquitous Service Management ARchiTecture. 49–53, 55, 56, 69, 73, 75, 91, 93, 95, 96, 101, 109, 124, 128, 130, 157, 163

UDDI Universal Description Discovery and Integration. 31, 32

UPnP Universal Plug and Play. 37

WSDL Web Services Description Language. 31, 38, 52

XML Extensible Markup Language. 31, 38, 40, 55, 128

1 Introduction

The worldwide population is ageing rapidly with an estimation of 1 in 5 people over 65 years old by 2030 compared to 1 in 10 today. Due to chronic aged-related illnesses, many progressively lose their autonomy and become more dependent on others, to finally reach the stage when they need round-the-clock care from their family members or caregivers. However, in the near future, the number of elderly people living alone in industrialized countries is expected to increase substantially. At the same time, and due to the recent economic downturn, health funding in most developed countries will most likely remain under pressure, and will not be able to grow significantly in the foreseeable future [1].

As a consequence, and in order to limit overall human and financial cost burden for the society, healthcare systems will have to find innovative ways to provide cost effective, but high quality care to an increasing number of elderly patients who are most likely to suffer from chronic disease or other disabilities. That is where smart connected homes and personal tele-health systems enter the game; they are recognized as key enablers to address the above mentioned challenges.

In this chapter, I will provide an overview of the different problematics resulting from the growing ageing society and the importance of Ambient Assistive Living (AAL) technologies to overcome these problems. Then I will present some examples of systems for ambient assistance and the main technical requirements they should fulfill. Finally I will position the work in this thesis as an approach for creating dynamic and adaptive framework able to handle uncertainty derived from technical problems or unclear human behavior in ageing environments.

1.1 High-Level Problematic

The rapid ageing of humanity is perhaps the most salient and dynamic aspect of modern demography. It implies an evolving impact on younger population and concerned countries. As a result, its influence on public health and national economies is becoming critical. In this section, I describe the principal factors leading to the ageing of the population.

1.1.1 Ageing Population and Dependency

According to Birren & Renner, ageing refers to *"the regular changes that occur in mature, genetically representative organisms living under representative environmental conditions as they advance in chronological age"* [2]. This definition represents the concept of ageing as distinct from disease and disability and links it to the processes discretely associated with the passage of time. However, different chronic diseases may appear, mainly during this period of life (e.g. heart disease, cancer, Alzheimer, etc.), which induce physical and cognitive disabilities of the ageing population and affect the dependency of this portion of humanity. Studies show that for the population aged 65 years and over, 9 out of 10 seniors reported having at least one chronic health condition and 4 out of 10 have at least three multiple conditions. Figure 1.1 represents some chronic diseases affecting the ageing population [3].

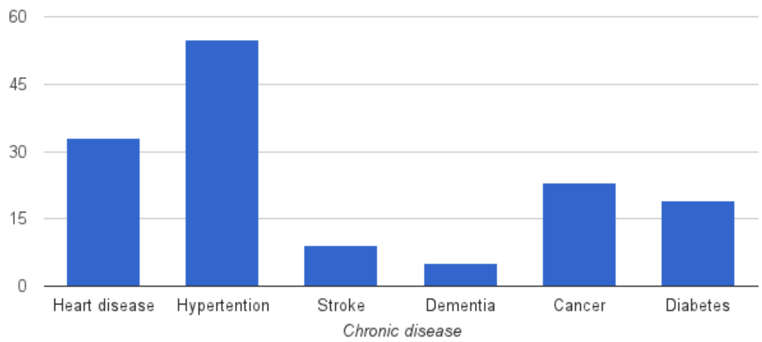


Figure 1.1: Chronic health conditions among the population age 65 and over

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 for some more than for others. 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. This difference is not only related to the age, it is also notable between people from the same age having different capabilities and limitations.

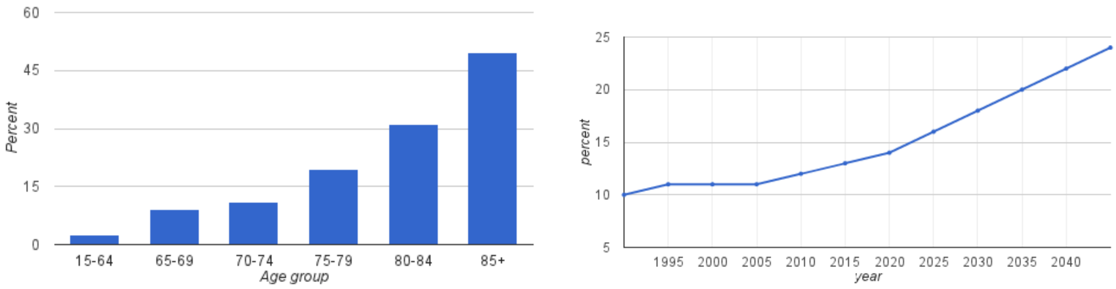
Based on the above presentation of ageing and related diseases, the process of ageing can be categorized in three main periods as follow:

- Primary Ageing: 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: or pathological ageing, concerns developmental changes related to lifestyle factors and diseases, such as cardiovascular disease or dementia, for the population 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: 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.

Global concerns about the ageing population have their origins related to the alarming increase in the number of dependent elderly and their care cost [4]. In fact, as shown in Figure 1.2a, 31.2% of people aged 80 to 84 years old and 49.5% of those over 85 years, need assistance to perform their Activities of Daily Living (ADL)s. It also estimates that about 10% of elderlies aged over 65 years old have cognitive deficits which affect their ability to perform ADLs. These numbers are continuously increasing. In 1990, 10% of the world population was dependant. We are today reaching 12%, and it is estimated to reach 20% in the next 20 years as shown in Figure 1.2b.

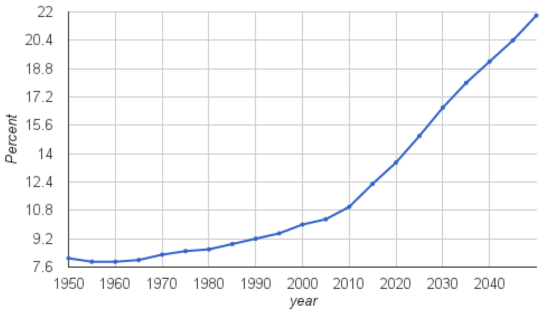


(a) Percentage of dependant people by group of age [5]

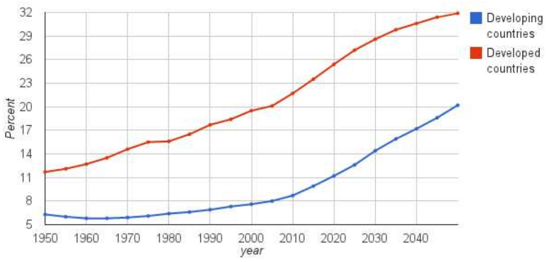
(b) Dependency ratio evolution over the years [6]

Figure 1.2: Statistics about dependant people

This increase is mainly related to the growth of the ageing population. In fact, Figure 1.3a exhibits that the world experienced only a modest increase in the share of people aged 60 and over during the past six decades, from 8% to 10%, but in the next four decades, this group is expected to reach 22% of the total population with a jump from 800 million to 2 billion people [7]. While this aspect of ageing population was, till the nineties, only related to developed countries, it is nowadays a global phenomenon, and it is accelerating, especially in the developing countries as it is observed in Figure 1.3b. In the developed countries, the share of ageing population over 60 years has increased from 12% in 1950 to 22% today and is expected to reach 32% (418 million) by 2050. In the developing countries, the share of those 60-plus has risen from 6% in 1950 to 9% today and is expected to reach 20% (1.6 billion) by 2050.



(a) World ageing population evolution [6]



(b) Ageing population comparison between developed and developing countries [6]

Figure 1.3: Statistics about ageing population in the world

Two main reasons could explain this explosion increase of the ageing population, the notable increase in life expectancy and the fertility rate decline.

- Increase of life expectancy/Decline in old age mortality: Globally, life expectancy increased by two decades between 1950 and 2010 (from 48 years in 1950 to 1955 to 68 years in 2005 to 2010) as can be extracted from Figure 1.4, and is expected to go up to 75 years by 2050. There are still considerable disparities between the developed countries, at 82 years, and the developing countries, at 74 years. However, this gap has narrowed greatly in the last decades. The life expectancy of older people has increased particularly rapidly; a person who reaches age 60 has more years of life left than in the past.

- Decline in fertility rate: As shown in Figure 1.5, the world's total fertility rate - that is, the number of children born per woman - fell from 5 children per woman in 1950 to roughly 2.5 today, and is projected to drop to about 2 by 2050. Most of this decline has occurred in the developing world, where the share of children in the population is expected, by 2050 to drop by half from the 1965 level.

  As families have fewer children, the older-age share of the population naturally increases.

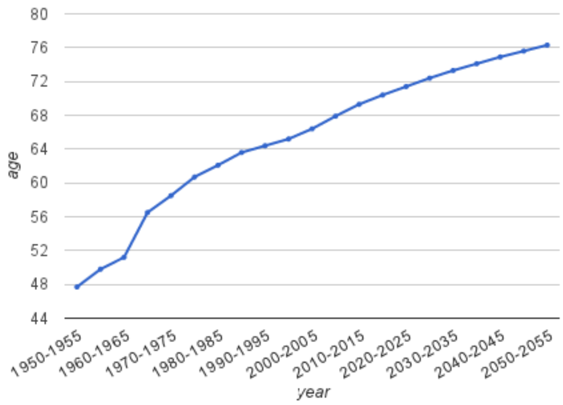


Figure 1.4: Life expectancy evolution [6]

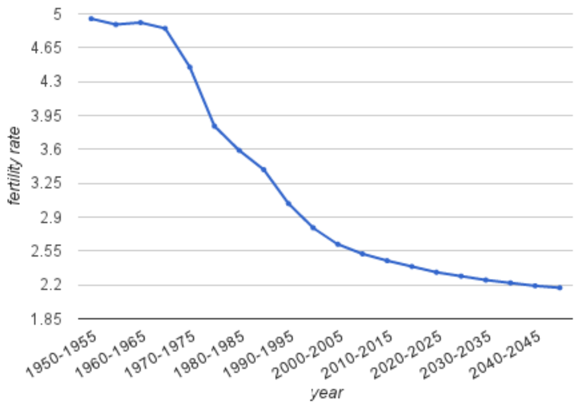


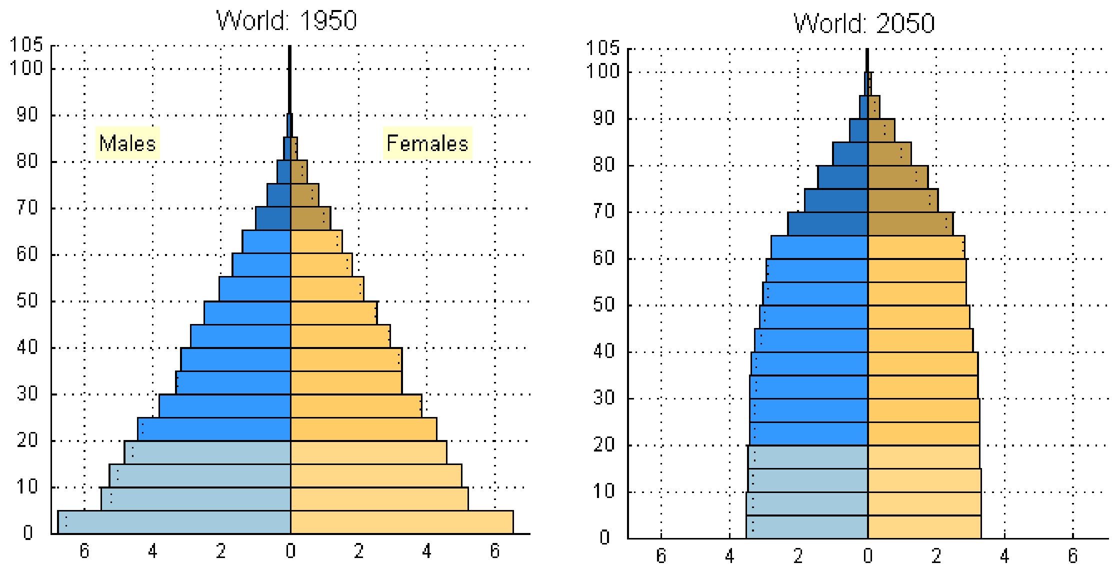
Figure 1.5: Fertility rate decrease [6]

Because of these reasons, the world demography is dramatically changing from a society with a young population in majority in 1950, as shown in Figure 1.6a, to a society with nearly a balanced share between the young and the old population in 2050, as can be noted in Figure 1.6b.

1.1.2 Cognitive Deficiency

Normal Ageing Cognitive Decline

One of the most annoying problems that upset ageing people is the decrease of their cognitive abilities. As reported by Bonder et al. [8] *changes in memory functioning with advancing age are expected, feared, and exaggerated in common folklore.* These changes concern different memory functionalities and affect mainly the working memory, the attention of the person and the capability to organize his activities. However some other abilities such as verbal ability, general knowledge and previous experiences remembering



(a) World demography in 1950 [6]

(b) World demography in 2050 [6]

Figure 1.6: World demographic change between 1950 and 2050

stay intact. While cognitive deficiency could affect different functionalities of the memory, it has an irregular impact on each of them.

- Sensory memory: also called *"primary memory"*, is the ability to hold large amounts of sensory information for a very short period of time and it includes a component for each of the five main senses (sight, hearing, taste, smell and touch). It allows individuals to retain impressions of sensory information after the original stimulus has ended. This functionality has a minimal age-related affection.

- Working memory: or the *"short-term memory"*, is the ability to hold information and manipulate it at the same time. It provides space for mental work in short duration with limited capacity. Short term memory is highly involved in activities of daily living with double tasking and information manipulation, such as listening then writing, listening then responding or watching then answering. This memory functionality is a strong evidence of age-related differences and considerably declines with age.

- Long-term memory: is a series of memory modules with each module responsible for holding different sorts of information for unlimited time. It is the memory functionality used to store everything you'll know forever.

Episodic memory: is the memory for specific episodes or occurrences and personal experienced events associated with time and place. This functionality decline with age, older adults have principally difficulties with new episodic memories such as the name of someone just met or the steps for using a new medical device.

Semantic memory: is a storage of factual information that accrues through a lifetime of learning and increases with time, it is the memory of meanings, understanding and other concept-based knowledge, such as the meaning of an apple or the meaning of liberty. This functionality is not really influenced by normal ageing.

Procedural memory: resides below the level of conscious awareness, it is automatically utilized for the execution of both cognitive and motor skills; from tying shoes to driving to reading. Procedural memories are accessed and used without the need for conscious control or attention; it is highly automatic and implicit. This functionality is well preserved and not influenced by ageing; however, it is slow and difficult to learn new procedures.

Prospective Memory: is the memory responsible of remembering to perform planned actions in the future. These actions range from simple tasks, such as calling a friend, to extremely dangerous situations, like taking medications. This functionality can be affected with ageing and it is influenced by the educational level and global intelligence of the individual.

Table 1.1 summarizes the main cognitive functionalities that can be affected by normal ageing. Different factors can explain the decrease of these abilities over the years, such as stress, anxiety, depression, inactivity, attention limitation and lack of organization in daily life.

Dementia as a Pathological Cognitive Decline

The evolution of cognitive functions decline could lead to a pathological condition of ageing people. A diagnostic evaluation for pathological cognitive decline includes mental status examinations, a review of the patient's past medical history and is completed by a clinical observation of the patient's symptoms, their beginning (sudden or gradual) and their progression over time. Mild cognitive impairment is a transitional stage between normal ageing and pathological cognitive decline. This condition can remain stable over the years; however it can improve or decline to pathological cognitive impairment.

Table 1.1: Cognitive functions decline in normal ageing

Memory functionality	**Ageing Effect**
Sensory Memory	Minimal
Working Memory	Fairly strong
Episodic Memory (new Info)	Fairly strong
Semantic Memory	Minimal some retrieval problem
Procedural Memory	Minimal
Prospective Memory	Mixed

Dementia is one of the most important cognitive related chronic diseases that affect the ageing population. Each year, 10 to 15% of the elderlies with mild cognitive impairment are diagnosed with dementia [9]. Dementia is affecting 5% of all people above 65 and over 40% of people over 85 years. The cost of care and assistance for this disease is getting very high; the World Alzheimer 2010 economic report [10] estimates that the worldwide cost of dementia is 604 billion US$ a year. It accounts for 4.1% of total disease burden among people aged over 60 years and 40% of people greater than 85. The number of people affected by this disease is increasing exponentially with an estimation of 35.6 million people with dementia in 2010 and the numbers are nearly doubling every 20 years [11]. During the next 20 years, the number of patients with dementia will increase by 40% in Europe, 63% in North America, 77% in the Southern Cone of Latin America and 89% in the developed Asia-Pacific countries. These figures are to be compared with the 117% growth in East Asia, 107% in South Asia, 134 - 146% in the rest of Latin America, and 125% in North Africa and the Middle East.

Dementia is a progressive, disabling, chronic disease, characterized by a progressive deterioration of intellectual and functional abilities, typically over a period of 7 - 10 years [12]. According to the Global Deterioration Scale (GDS) [13], cognitive and functional abilities are categorized into 7 stages, ranging from no cognitive decline in the first stage to very severe cognitive decline in the seventh stage. These stages are described in Appendix A. Dementia is classified into 5 stages (3 - 7) according to the GDS. At stage 3, the symptoms can be subtle and the patient can live independently without assistance [14].

During stages 3 to 5, patients suffer progressive cognitive decline and experience increasing difficulties in performing ADLs [15]. 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 learnt tasks (apraxia). This means that the caregivers have to be present to support patients during their activities and slowly, over time, need to increase this support they provide as the disease evolves. Caregivers need to remain informed on how patients are performing their ADLs and to provide support as and when appropriate. They need to provide just the right amount of assistance so as not to take over the task but still allow the patient to retain some level of independence [16]. Over time, as patients need more help, caregivers also experience increasing levels of stress and burden. Care-giving for a dementia patient can be physically and emotionally demanding and has been found to be more stressful than care-giving for older people suffering from other ailments [17].

This situation has motivated computer researchers to look for solutions to solve the problem, taking advantage of the developments in information technologies and tools for identification and monitoring. The idea is to design intelligent systems that are able to support demented persons in performing their ADLs, and to provide all commodities necessary to maintain/increase their autonomy. The guiding principle is to provide remote monitoring of demented persons as they carry out their daily tasks and to intervene when appropriate.

1.2 Gerontechnology

Gerontechnology is a composite of two words, *"gerontology"* and *"technology"*. Gerontology is the scientific study of ageing concerned with research on its biological, psychological, social, and medical aspects. Technology is interested in research, development and design of new and improved products and services. As described above, the ageing population is emancipating and its importance in society is growing, which lead to different problems with the age related changes and pathological situations of elderlies. Technology is also spreading, especially with the appearance of new materials and Information and Communication Technology (ICT). Unfortunately, these two evolving processes tend to be developed while unconnected one from another, which led to the emergence of studies on gerontechnology. Gerontechnology is an interdisciplinary domain that aims at directing technology towards the ambitions of seniors such as good health, independent living, and full social participation. It allows the design and development in the engineering disciplines based on scientific knowledge about the ageing process [18].

Research studies in gerontechnology have shown the effectiveness of technology in preventing age-related diseases and losses in strength, endurance and other physical or cognitive abilities. Technology also compensates with the decline of these capacities, through different products and techniques such as reading glasses to compensate for diminishing flexibility of the eye's lens. Furthermore, for caregivers working with less able older persons, gerontechnology provides technical support like technologies for lifting and transferring people who are incapable of moving alone. Another aspect where gerontechnology can be beneficial for senior citizens is in enhancing their performances and opportunities in new roles and situations such as performing new leisure and living activities, or adapting themselves to new social situations.

The integration of these technologies in ageing people environment and the Quality of Life (QoL) improvement of older citizens with the use of Assistive Technology (AT) in their daily life is a crucial topic that have been tackled in studies on Intelligent Environments (IE) and AAL. Some of the aspects that need to be handled are the ease of use and adaptability of these technologies for ageing use and their deployment as ubiquitous and unperceived tools with the minimal possible interaction.

1.3 Intelligent Environments

Intelligent environment or smart space is the vision of including smart technologies and ICT into the computation of environment, creating a space that brings computation into the physical world. Types of IEs range from private to public and from fixed to mobile, such as health centres, airports, cars, Assistive Environment (AE)s, smart homes and smart buildings. Elaborating such environments requires expertise in different enabling technology fields, essentially microelectronics (power consumption, sensors manufacturing), communication and networking technologies (broadband and wireless networks) and intelligent agents (context awareness and ontologies).

1.3.1 Ambient Intelligence

Ambient Intelligence (AmI) is a derivation from IE definition and combines both IEs or smart spaces and the incorporation of ubiquitous computing to provide pervasive spaces. This vision was first proposed by Mark Weiser [19] in 1991, when computers had the size of a room and the idea of a computer being camouflaged within any environment was an unimaginable notion. Nowadays, his vision has undoubtedly come possible. Different appliances have successfully become integrated in our surroundings to such an extent that we use them without being even conscious of their existence. As the computer disappears in the environments surrounding our activities, the objects are augmented with ICT components (i.e. sensors, processors, actuators, memories, wireless communication modules) and can receive, process, store and transmit information. Space also undergoes a change, towards becoming augmented AmI space embedding sensing, processing, actuating and networking infrastructure in order to offer a set of services in the digital AmI space. By using the ambient technology, the AmI spaces support user tasks and allow people to carry out novel or traditional tasks in unobtrusive and effective ways [20].

In that sense, an ambient intelligent space is a pervasive, transparent infrastructure able to observe people without interfering with their life and then reacts to their needs and requirements. Thus, there is an explicit focus on the individual in the environment. Indeed, with AmI, the vision is changing from an individual who needs to adapt to what the technology can provide into an environment and technology that must adapt to the individual. In the concept of AmI, the environment is mainly composed of devices, applications, services and their interfaces and of sensing systems that enable it. Using these systems, the environment becomes sensitive to the presence and the activities of the persons in it and can react to their needs and requirements.

The five main features that an ambient intelligent system should fulfill with respect to its relation with users can be characterized by [21]:

- Non-obtrusive: the system should not interfere with the user's life style through invisible, embedded and distributed devices.
- Context aware: the system should be able to recognize and anticipate the context the user is evolving in and use this context to react to his needs and requirements.
- Personalized: users' profiles and environments are heterogeneous. The system should

be customizable in order to fit in different environments and to be used by different users.

- Adaptive: the system's behavior can change in response to the actions and the needs of a person. New services and functionalities can be added with respect to the user needs.

- Anticipatory: it anticipates the person's desires and environment status; it is context predictive and proactively enabler.

1.3.2 Ambient Assisted Living

AT could be defined as any tool, equipment, system, or service designed to help develop, maintain or improve a person with a disability in all aspects of his/her life. It helps people of all ages who may have a broad range of disabilities or limitations. Ordinary environments are not adapted for ageing people having lost the ability to perform their ADLs such as opening a door, eating, or even showering. A physical space gathering one or many ageing people with dependency problems and their needed ATs, called AE, should be able to provide users with accessible services and activities they want to perform using existing ATs and emerging technologies.

Ambient Assistive Living represents an emerging application area where AmI solutions based on context aware, plastic interface, configurable human environment interaction, as well as probabilistic and rule based reasoning can play an important role to solve the ageing problems raised earlier. AAL can enhance ageing in place by helping elderly people with their ADLs. The concept of AAL consists of a set of ATs, targeting the extension of the time that elderly and disabled people live independently in their preferred environment, i.e. their own home, neighborhood and town where they are used to live. It covers personalized home care assistance, smart homes, assistive cities, etc. ATs provide personalized continuity of care and assistance, dynamically adapted to the individual needs and circumstances of the users throughout their lives [20]. They are targeted at improving the organization of healthcare providers, improving therapy and rehabilitation, and enhancing prevention and care. An overview of AAL with the different interacting domains is represented in Appendix B.

ATs could also be used for the remote care of elderly and dependant people. They serve to provide information for caregivers about their relative patients situation or send alerts in case of detected risk. This type of assistance is called “Telecare”. The main difference between AAL and Telecare lies in the fact that AAL systems aim to provide support for the patients themselves and encourage them to keep their independence in performing their activities. In the other hand, Telecare aims more to keep remote caregivers informed about their patients situation, asking them to interfere in case of problem without a direct contact or interaction with the patient. Different studies have already been conducted in the domain of Telecare [22, 23] in order to identify trigger factors leading older people to need support and the type of Telecare services that can be provided. Most of the systems for Telecare are related to vital signs monitoring and telephone follow up by nurses [24].

In this thesis, the context of my work is primarily related to smart home technologies in order to build an AAL environment for ageing with cognitive problems. It is also related to Telecare in some way as these technologies can trigger caregivers when the patient is not reacting to the assistance provided. In the next paragraph, I will present some existing solutions of smart homes interested in this area of research.

1.4 Ambient Assistive Living Systems for Cognitive Decline

Ambient Assistive Living technologies can be used to assist people with cognitive decline and their caregivers. Today, they are used in diverse healthcare applications and they are expected to increase efficacy and efficiency of healthcare providers [25, 26]. They are targeted at improving the organization of healthcare providers, improving therapy and rehabilitation, and enhancing prevention and care [26].

1.4.1 Gator Tech Smart House

The Gator Tech Smart House [27] is an experimental laboratory and an actual live-in trial environment for validating technologies and systems developed in the Mobile and Pervasive Computing Laboratory of the University of Florida. It is mainly dedicated to carry out experiments for applications for elderly and disabled persons. The house has been used as a platform to implement, test and validate various systems and human centered applications. It is one of the few research facilities in the US where actual human subjects are engaged in the research by living in the house for variable periods of time. The Mobile and Pervasive Computing Laboratory is developing programmable pervasive spaces in which a smart space exists as both a runtime environment and a software library.

Service discovery and gateway protocols automatically integrate system components using generic middleware that maintains a service definition for each sensor and actuator in the space. Programmers assemble services into composite applications, which third parties can easily implement or extend. The project's goal is to create AEs such as homes that can sense the context of their residents and enact mappings between the physical world, remote monitoring and intervention services [28].

1.4.2 SIMBAD System

The SIMBAD system (Smart Inactivity Monitor using Array-Based Detectors) is a system that detects falls of the elderly since this problem is a major health hazard for them and a major obstacle to independent living. It uses a low-cost array of infrared detectors and does not need wearable devices. It employs a neural network to classify falls using vertical-velocity estimates derived directly from the infrared sensor data [29]. A field trial and user research indicates that SIMBAD could significantly enhance the functionality and effectiveness of existing monitoring systems and community alarm systems.

1.4.3 CSCC System

Computer-Supported Coordinated Care system (CSCC system) [30] uses technology to aid the network of people who support an ageing person living at home. The authors conducted interviews with people involved in the care of elders to identify their needs and subsequently conducted an in-situ evaluation of a technology probe to study how a CSCC system might help satisfy these needs. The results of these evaluations are used to identify challenges faced by people caring for seniors and offer guidelines for designers of coordinated care technologies.

1.4.4 ISISEMD Project

The Intelligent System for Independent living and SElf-care of seniors with cognitive problems or Mild Dementia [31] supports the independent living of elderly people in general and in particular the group of elderlies with cognitive problems or mild dementia. It supports at the same time the formal and informal caregivers in their daily interaction with the elderlies. ISISEMD provides several services that improve the elderly ability for self-care by supporting their basic daily activities in a way that prevents health risks in their homes. The services will also strengthen the daily interaction with their social sphere - partners and relatives, friends and care-givers, giving them the feeling of safety and preventing their social isolation. Ageing cognitive training and activation are also strengthened. The system contains 3 different service bundles (basic services, intermediate and high level) that allow the escalation of the service provided to the end-users based on their needs, providing different pricing schemes [32].

1.4.5 ROSETTA Project

The ROSETTA project [33] aims at developing technologies to help persons with dementia and/or Parkinson's disease to live independently longer. It consists mainly of three software packages on one ICT infrastructure. These three software packages offer the following services:

- Day navigator: provides a memory support for persons with memory problems. A touch screen is placed in the patient's home, on which a diary is made visible providing memory support. The screen also gives access to a photo phone and a photo album.

- Early Detection System: it is difficult for care providers to see slow changes occurring in the daily life pattern of persons living alone. In order to keep caregivers up-to-date and tailored, a software is monitoring the daily life pattern of the patients. Care providers as well as informal carers living elsewhere can monitor the daily life pattern via a computer program. Notable changes in the daily life pattern are reported.

- Unattended Autonomous Surveillance: allows an automatic surveillance for persons with advanced dementia or persons with a combination of memory problems/mild dementia and Parkinson's disease, without the need to wear technological equipments. Possible emergency incidents are detected by sensors and a camera and then reported to a mobile care team. The care providers in this team respond to the reporting.

Other prototypes were developed for specific scenarios to assist patients during different stages of cognitive decline ranging from healthy ageing to severe cognitive impairment [34, 35, 36]. Remote monitoring systems are used for mobility measurement to estimate disturbances in motor activity of the patients to prevent risks of accident [37]. Some systems use video and audio recording for patient tracking in order to analyze their activities

[36, 38]. Orwat et al. [26] make a literature review of systems developed for AAL in health care classified by country and showing the status they have reached (prototype or pilot testing, clinical or medical trials, and regular operation) and the intended stakeholders.

1.5 Requirements of Ambient Assisted Living Systems

Different technical requirements emerge from the deployment of AT for ambient assistance. These requirements need to be met in order for the proposed solutions to be accepted and utilized in real life.

1.5.1 Context Awareness

For an AT to work, obviously it must be able to sense and manage context information. However, the term *"context"* can imply an extremely wide range of concepts. Abowd et al. [39] define context as *"Any information that can be used to characterize 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 application themselves."* They proposed the notions of primary context (localization, identity, activity and time) and of secondary context, where the secondary context is deduced from the primary context and may be used to make decisions at a higher level of abstraction.

The following are some commonly considered context information that an AT could provide and manage [40]:

- Identity of the user
- Spatial information: location, orientation, speed, acceleration, objects relationship in physical space, etc.
- Temporal information: time of the day, date, season of the year, etc.
- Environmental information: temperature, humidity, air quality, light, noise pattern and level, etc.
- Social situation: who the user is with, the nearby people, family relationships, people that are accessible, etc.
- Nearby Resources: accessible devices, hosts, other facilities, etc.

- Resource usability: battery capacity, display resolution, network connectivity, communication bandwidth and cost, etc.
- Physiological measurements: blood pressure, heart rate, respiration rate, muscle activities, tone of voice, etc.
- User's physical activity: talking, reading, walking, running, driving a car, etc.
- User's emotional status: preferences, mood, focus of attention, etc.
- Schedules and agendas, conventional rules, policies, etc.

1.5.2 Privacy

Guaranteeing the respect of patients' and caregivers' privacy and the confidentiality of the research records is a major requirement. It is a crucial characteristic that has an impact on the acceptance and usability of these technologies by the end-users. Many systems use video and audio recording to provide assistive services [38, 36]; this is however a serious issue when targeting real life deployment for AAL. Therefore, AT should be committed to use only non-intrusive sensors, without resorting to any video or audio recording. In addition, collected data should be anonymized and personal information should be removed. The communication of this information, if needed, should also be secured and anonymized.

1.5.3 Multiple People Management

Another aspect that needs to be taken into account in the deployment of AT is the fact that patients are not living alone. It is important to identify people in the environment when aiming at providing personalized assistance and taking into account the patient's profile. An identification mechanism needs to be established to identify people in their environment. Some non-intrusive solutions can be used to achieve this requirement. Technologies such as Radio-Frequency IDentification (RFID) tags for the identification of patients have been widely used [41, 42]. These passive tags can be embedded inside plastic bracelets worn on patients' wrist while RFID readers are placed at major points of interest in the patient's environment. Systems can also use reasoning and inference to infer detected people's identity if such information is missing.

1.5.4 Design for failure

Ambient systems are prone to crashes. This is partly due to the reliance on several wireless communication protocols. Among others, thick walls, temporary interferences or poor coverage are factors that affect the system's reliability. Some other problems are related to the sensors usage, being a battery issue or simply a curious user pulling them off from his environment. When systems are deployed in real settings, there is no more constant supervision by technicians while actual users count on the provided services to maintain their safety. It is becoming very costly to analyze a crash or restart the system as it requires a specifically skilled person to go down to the deployment site. Moreover, if a crash is not detected or crash reports are not sent in a timely manner, this could question the safety of end-users. Hence, the stability issue takes a whole new dimension in real deployments. *Designing for failure*, which is a nice and luxurious feature for laboratory prototypes, becomes a necessity in real-world deployments. It is based on the idea that crashes are inevitable and should be taken into account during the design and implementation phases. It encompasses the following processes: ensuring the automatic system recovery after crashes, proper logging of information for crashes analysis, and proper signage about the system health for end-users.

1.5.5 Dynamism and Adaptability of the system

The dynamism and adaptability of the system is an important issue that needs to be tackled in order to be able to deploy assistive systems in different environments and to take into account different users' profiles or even the progression of their diseases. Indeed, different patients have different behaviors and profiles, with the possibility of progression of their disease. This will have an influence on the choice of interaction devices and assistive services to be provided for each patient. In that sense, and unlike many systems who have only one specific service (or functionality) to provide [36, 34, 35], the proposed software platform should be designed to be dynamic and scalable, allowing the integration of new assistive services depending on the needs of each patient.

Integrating new assistive services in the platform possibly implies adding new sensors and interaction devices in the environment. The dynamic aspect of the framework helps to include them at runtime. Targeting a large scale deployment also relies on the dynamism and adaptability aspects of such systems. Indeed, it introduces more diversity in patients' profiles and needs that should be handled, as well as more heterogeneity of the environments in which sensors, interaction devices and assistive services will be deployed. The system has to be designed so as to improve maintenance efficiency and upgrade speed by simplifying the associated technical process.

1.5.6 Handling Uncertainty

The main peculiarities of context information lie in its imperfection. Inconsistent or incomplete information are common due to faulty hardware, delays between production and consumption of the information, or even networking problems. Assistive systems could also lack knowledge about the context in order to reason and take decisions. This leads to an uncertainty in data management and decision making.

The notion of uncertainty can be very risky and unsafe when it deals with ambient assistance. It leads to imprecise decisions that can affect the supervised person or can even obstruct the decision process and the provision of assistive services. A mechanism that handles uncertainty during the reasoning process needs to be established in order to cope with this kind of deficiencies.

1.6 Thesis Positioning

In this thesis, I present our vision for building a framework for AAL based on technical requirements gathered from our on-site observations and focus group discussions. This work was motivated by our goal for large scale deployment of AAL solutions which has raised crucial needs that should be tackled related to the dynamism and adaptability of the proposed system and the management of uncertain information for context awareness.

Existing systems, such as the solutions presented earlier, are providing a set of defined services and scenarios. Most of them are also limited at prototyping and laboratory testing [25]. Our idea is to go beyond this for a real world deployment. This allows the identification of more concrete requirements and problems that need to be addressed and which cannot be identified in laboratory testing. Issues, met neither in the design phases, nor during laboratory testing, will appear and affect the usability of the proposed solutions. In fact, most studies in the field of smart homes and dementia assistance work perfectly in a laboratory testing environment. However, they fall short when they tend for commercialization or real deployment due to the lack of collaboration with professionals in the domain and the restriction of these studies to laboratories prototyping and testing. Our idea was not only to develop a solution and deploy it in real world for validation. The deployment allows to identify concrete needs that are not integrated in the initial solution. Therefore, the purpose of the real world deployment was to have an iterative needs evaluation and to evolve the system design based on identified needs as shown in Figure 1.7. This allows more integration of the medical staff in the design and evaluation of the system. We envision that such a multidisciplinary design approach, supporting a deployment in real life settings is crucial; and that a simple system developed and validated in this way is more relevant and valuable than a well-featured solution proven to be stable in a laboratory.

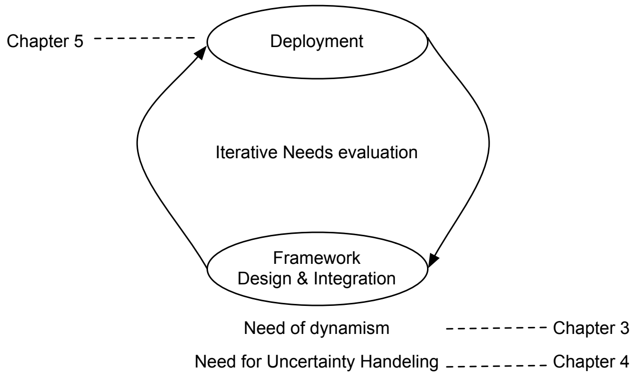


Figure 1.7: Deployment and system integration interaction

Challenges related to the heterogeneity of the environments in which sensors, devices and services will be deployed, as well as the diversity of patients' needs and profiles are major constraints that could object the spread of the use of AAL solutions. AAL environments are usually evolving by the introduction or the disappearance of sensors, devices and services to respond to changes in patients' conditions. In addition, each patient has a specific profile that influences the choice of interaction devices and requires particular assistive actions. The selection of required assistive actions affects the decision on the set of sensors that need to be installed. Therefore, a generic framework that is able to adapt to such dynamic environments and to integrate new sensors, devices and services at runtime is required.

In addition, uncertainty is one of the principle problems I have deduced from our work on the field. It is related to technical problems emerging from sensors inefficiency or network problems that induce uncertainty about information and events received by the framework. Uncertainty is also related to human behavior, in some cases, the situation of the assisted person is imprecise and the system is not able to classify this behavior. A notion of uncertain representation and decision should be introduced to reply to these needs.

Our main contribution in this thesis consists in providing an approach based on semantic representation and the Service Oriented Approach (SOA) that guarantee the dynamism and adaptability of the proposed solution, taking into account uncertain information derived from technical problems or unclear human behaviors.

1.7 Thesis Outline

This thesis is structured in five chapters. In the first chapter, I have presented the background of the thesis, showing the need of ambient assistive living to deal with the rising problem of ageing population and cognitive related chronic diseases. I have also exhibited some existing solutions for AAL, and diverse technical requirements that have emerged from our on-site deployment. I have ended this chapter by positioning the thesis objective to handle principle requirements we judge fundamental for the success of the proposed solutions and their acceptance by the end-users. These requirements are mainly the dynamism and the extendibility of the proposed framework and the management of uncertainty. In the rest of the thesis, I will at first focus on the technical part showing our approach for handling the requirements raised from the deployment (chapters 3 and 4), then I will present the deployment approach and the main results obtained (chapter 5). In the second chapter, I provide an overview of the existing work ensuring the dynamism of AAL systems and their relation with our vision, detailing most of the key approaches and technologies that could be used to guarantee the dynamism. This chapter ends with an opening on uncertainty handling which emerges from the discussion about dynamism. Some related work dealing with uncertainty is provided. The third chapter details the proposed framework and the transition from a static configuration to a dynamic and extendible configuration based on my proposed approach of semantic Plug&Play. A validation of the proposed approach concludes this chapter. The fourth chapter is dedicated to a discussion about methods of uncertainty measurement and representation, and for reasoning under uncertainty. I detail my approach and implementation for dealing with uncertainty and how it was integrated in the proposed framework. I end the chapter with a validation of the proposed approach. In the last chapter, I give an overview of our deployment experience, with the deployment approach used and the different phases of our trial. I provide results from the real world deployment and qualitative feedback from caregivers and doctors.

2

The State of the Art

2.1 Introduction

Different approaches can be explored and adopted in order to support our vision for the creation of a dynamic and extendible AAL solution that can adapt to the diversity of deployment environments and to the evolution of end-users diseases. At the infrastructure level, a modular approach can be very beneficial compared to the monolithic approach. The modular approach is supported by the declarative approach through the use of semantic modeling and reasoning allowing more separation between the functional and descriptive aspects of the system. Such a dynamic AT faces multiple problems related to the uncertainty about the events received from sensors and the absence or lack of context information, therefore another level dealing with uncertainty should be introduced.

In this chapter, the main approaches supporting the dynamism in AAL environments are presented with the different technologies that can be used to implement it. The chapter starts with an overview of existing research works focusing on this problematic. After that, different approaches and technologies for service modeling and interoperability are introduced to explain our choice for the modular approach. The modular approach is supported by the declarative approach through the knowledge modeling and reasoning, presented in the second part of the chapter, in order to guarantee the dynamic aspect. An opening on uncertainty emerges at the end of this state of the art. A discussion about the selected approaches and technologies concludes this chapter.

2.2 Overview of AAL Dynamism

Some work has contributed to the dynamism and adaptability of systems in pervasive spaces and AAL environments. Existing research on application polymorphism [43] has helped on the portability of applications through different devices. It enables the application to modify its structure in order to adapt to the device on which it will be used. This approach is based on the decomposition of the application into smaller components that can be independently adapted and recomposed to obtain a semantically similar application on the specified device. This method is considered as complementary to our proposed approach, where I focus mainly on the detection of new devices and their integration into the semantic and reasoning model for the devices selection process. Once the assistive service to be provided and device of interaction are selected, the application polymorphism approach is used to adapt the assistive service to the selected device. Another approach is focusing on the network selection during device mobility [44]. The network selection in this work is based on the user's profile and on services relevance and importance in addition to the network parameters. This leads to select the most appropriate network for the user in order to be provided by interesting services. Once the network is selected, this method matches with our approach where I integrate the new detected devices into the framework, select the appropriate service and provide it on the most adequate device. Some work already exists allowing to AAL solutions the discovery, and the interaction with devices [45, 46]. However, no semantic bindings are added to the devices in these approaches; thus, either they do not support any device selection or they are only based on the devices intrinsic characteristics without reasoning on the context. On the sensor part, semantic reasoning has been used in wide-area sensor networks to perform the sensor network self-configuration [47]. However, there is not much work on sensors' Plug&Play and on the semantic binding of newly discovered sensors.

In a similar work supporting dynamic platforms [48], integrated devices and sensors are preconfigured and predefined for specific functionalities; thus, provided services are only based on one sensor and the platform cannot provide assistive services based on multiple sensors. Our approach is to bind assistive services to multiple context information. Hence our proposed framework authorizes the discovery and integration of different sensors and interaction devices, and then attaches them to several assistive services. It also allows their reconfiguration, when needed, to be used for other assistive services. The dynamic integration of entities based on the use of a middleware and some semantic representation is introduced in a position paper by Helal et al [49]. I have followed a similar approach based on the use of the modular and declarative approaches with the use of the Device Profile for Web Services (DPWS) communication mechanism.

2.3 Service Model

2.3.1 Modular Approach vs. Monolithic Approach

The modular approach is a top-down design approach that emphasizes the separation of a system functionalities into independent, loosely coupled and interchangeable modules. Each module contains everything that is necessary to provide a defined function or a set of similar functions. The coordination between different modules of a system is insured via *interfaces*. These interfaces are the signature of the different modules and they specify the functionalities provided by each of them for the rest of the system. Other modules do not need to be aware of the working code (implementation) corresponding to the functionalities declared in the interfaces. This design approach is very beneficial for AAL solutions. In fact, as complex systems, the modular approach enforces their logical structure by breaking their complexity into simpler tasks making them more efficient and easier to understand and modify. It is possible to study an AAL solution while exploring its modules one by one at a time. The whole system can therefore be better designed as it is better understood. Moreover, designing and developing an AT solution involve different domains (sensing, network, reasoning, data mining, Human-Machine Interaction). Adopting such an approach allows multidisciplinary players to work on several parts of the system at the same time with a little need for communication, thus making the development of the system faster. The modular approach is an important factor when targeting the dynamism and adaptability of AT systems as it improves the flexibility of the proposed solutions. With the modular approach, it is possible to make drastic changes to one module without the need to change the other modules. In fact, modules are substitutable and reusable. A module can replace another at design time or even at run-time without reassembling the whole system. It can also be integrated into another system to provide its functionalities.

2.3.2 The Service Oriented Approach

The Service Oriented Approach SOA can be seen as a continuation of the modular approach. It represents a set of principles and methodologies for designing and developing systems in the form of interoperable services delivered and used on demand [50]. SOA defines three main elements as shown in Figure 2.1

- Service provider: it is the entity which is implemented a service specification or description.
- Service requester: also called consumer, is the entity client that invokes a service provider to use a service. This can be a user application or another service.
- Service registry: a software entity that acts as a service locator where new services are published and delivered. It implements the discovery mechanism and suggests service providers for the requester of a specific service.

These elements play different roles which define the contracts between them as follow:

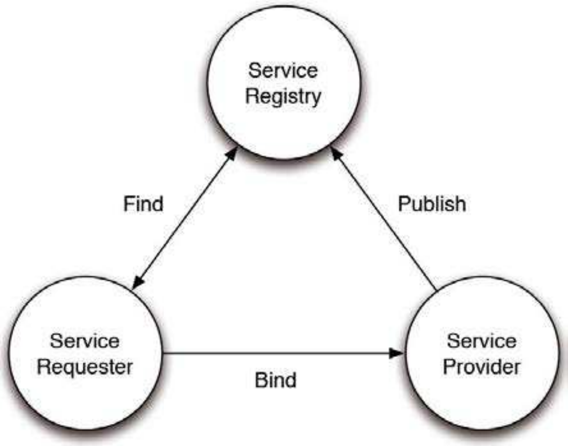


Figure 2.1: SOA basic architecture

- Publish: is an operation that acts as service registration or advertisement. It operates between the service registry and service provider. The service provider publishes a service in the service registry.
- Find (Discover): it is the contract between a service requester and a service registry. The Find operation is executed on the registry according to a search criteria specified by the requester. Search criteria may be the type of service, QoS (Quality of Service), etc... The service requester looks for a specific service in the service registry. The service registry replies with the identifications of service providers that provide the requested service.
- Bind: this operation binds both the service provider and requester in a client/server-like relationship. The service requester invokes the service from the service provider.

The SOA approach appears to be a convenient architectural style towards meeting one of the key objectives of this thesis, that is, to guarantee the dynamism of AAL solutions in order to adapt to the environmental changes and users' needs. In fact, SOA allows structuring the system in a modular way, permitting its flexibility, scalability, reconfigurability, and ease of replication. In addition, SOA has been accepted as a mature approach that solves the problems of interoperability between different technologies and the ability to add functionality without having to redo systems [51]. This is achievable by exploiting the services collaboration principle which helps to facilitate the communication and information exchange between an AAL platform and the different entities in the environment (sensors, actuators and devices) hosting different technologies. The discovery principle in SOA is very substantial enabling the discovery and interpretation of other services. This aspect is relevant to AAL platforms allowing them to respond to the environment changes and dynamism by the discovery of newly integrated entities in the environment. The principle of loose coupling also helps in the system dynamism and adaptability by facilitating the integration of newly discovered entities functionalities without recoding or interrupting the system. In fact, unlike other programming paradigms, such as the object oriented programming, SOA loose coupling aspect contributes in the system dynamism to respond to the environment changes by integrating/removing modules representing discovered/removed entities at runtime without affecting the system functionality.

Other SOA aspects such as service contract, encapsulation and abstraction help in managing environment entities complexity. They also allow the extension of the AAL middleware functionality to include newly integrated modules in the system. This ensures the use of all available resources in the environment for better assistance of the patient [52].

Different specifications and implementations have been realized in order to bring out the service oriented approach into practice. Hereafter are presented the two most predominant of them: the web service and the Open Service Gateway initiative (OSGi)

Web Service

The definition of Web Service encompasses different systems; however the usage of the term commonly refers to those services based on standardized techniques. Web services are based on the use of Simple Object Access Protocol (SOAP)-formatted Extensible Markup Language (XML) envelopes, have their interfaces described by Web Services Description Language (WSDL), and use Universal Description Discovery and Integration (UDDI) to locate and discover other services interfaces, as well as to subscribe to this registry. With web service, services are accessible over standard Internet protocols independently of platforms and programming languages.

The three main elements described in SOA are reproduced in the web service based systems:

- Service provider: it is the responsible for creating web services and publishing their WSDL description and access information to the service registry. The service provider decides on which services to expose to the UDDI registry and to which category they will be affected for a given service broker.

- d
  g
  s
- d
  a
  a

F e
provi

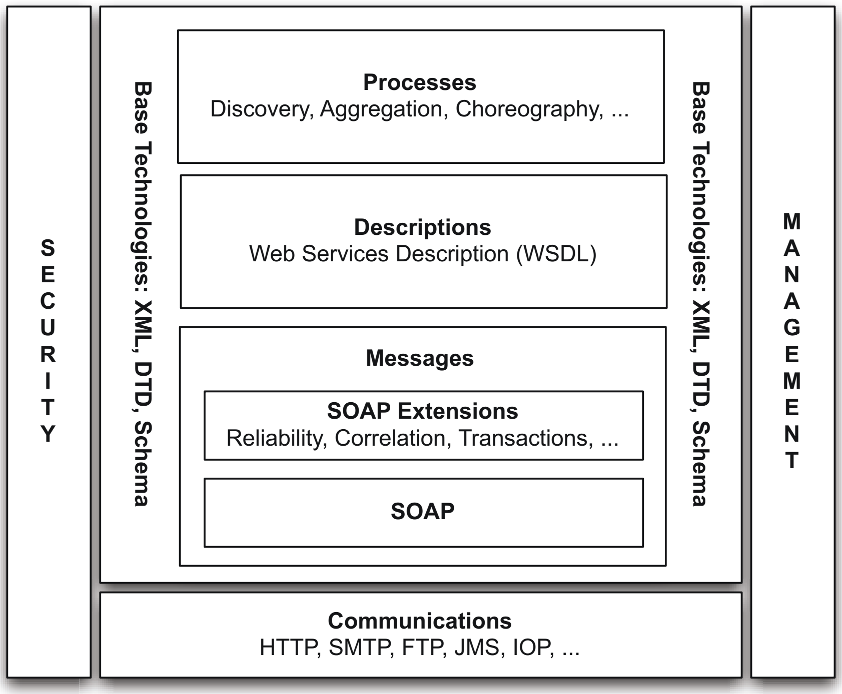


Figure 2.2: Web service architecture stack

The Open Service Gateway Initiative

The OSGi[2] framework is known as a standardized module system for networked services that is the foundation of enhanced service oriented architecture. Its standards are defined

[1] http://www.w3.org/TR/WD-ws-arch-20030808/
[2] http://www.osgi.org/

in the OSGi Alliance and published in the OSGi specification documents.

The scopes of the OSGi Service Platform are as follows [53]:

- Providing a standard, non-proprietary, software component framework for manufacturers, service providers, and developers. The fact that the OSGi specifications are an open standard enables a fair playing field for all participants.
- A cooperative model where applications can dynamically discover and use services provided by other applications running inside the same OSGi Service Platform. This cooperative service model is considered as a key element for service dependencies.
- A flexible remote management architecture that allows platform operators (the organization that manages the platform) and enterprises to manage thousands or even millions of Service Platforms from a single management domain.

An OSGi framework is basically a *container* running functional components called in OSGi terminology *bundles*. Life cycle management is one of the most prominent features of the OSGi framework. It provides the necessary mechanisms to allow remote management of bundles and also allows bundles to manage other bundles life cycles. Using these mechanisms and based on the OSGi dynamic component model, bundles can be remotely installed, started, stopped, updated and uninstalled at runtime without affecting other bundles or restarting the framework. The OSGi dynamic service registry allows bundles to register, listen and detect the addition or removal of services, and adapt accordingly [54]. The different layers of the OSGi framework are represented in Figure 2.3.

Thanks to its modular development approach, OSGi finds wide usage in the area of auto-mobiles, Mobile/PDA, industrial automation, building automation, Smart Home and E-Health application development and has been used in numerous projects for smart homes development. The OSIRIS project [55] defines an across-domain open source service platform based on OSGi that will provides support for service provisioning, aggregation and delivery. It also defines a mechanism for remote service invocation between several OSGi platforms. A smart home architecture for heterogeneous network [56] was also proposed using the OSGi gateway allowing the integration of services and devices from different domain. The SIRENA project [57] proposes a similar technology for device integration in heterogeneous domains. The architecture is based on the OSGi and the device profile for web services. In the context awareness domain, OSGi has been used as an infrastructure for context-aware applications in [58] in order to support context acquisition, discovery,

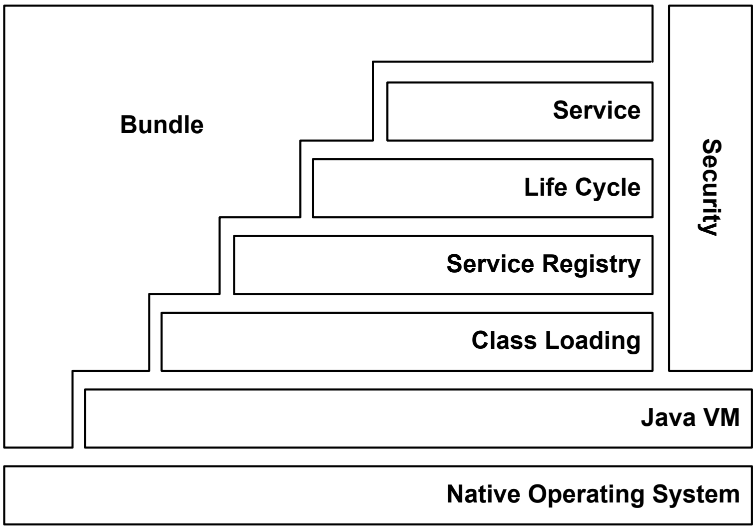


Figure 2.3: OSGi framework layers

and reasoning and adapt to the changing contexts in dynamic environments. A similar work has been realized by Ricquebourg et al. [59] with a network infrastructure to convey information emitted by heterogeneous smart objects and an OSGi software architecture to manage information and to provide the more adapted service by the way of heterogeneous sensors. The OSGi gateway serves as the central coordination point for managing the home network, spanning multiple heterogeneous communication technologies.

There are numerous open source implementations of the OSGi specification such as Concierge, Knopflerfish, Equinox and Apache Felix.

Concierge Concierge[3] is an optimized OSGi framework implementation with a file footprint of about 80 kBytes. This makes it ideal for mobile or embedded devices; however, this framework does not implement the OSGi Specification Release 4 (R4) which obstructs it from communicating with other containers running on other devices.

Knopflerfish Knopflerfish[4] provides an implementation of the OSGi specification R4. It's an easy-to-use open-source framework with GUI support for a graphical overview of the OSGi framework, though it is not adequate for small devices integration with a size of 50 MBytes at first use.

[3] http://concierge.sourceforge.net/
[4] www.knopflerfish.org

Equinox Another framework that implements the OSGi specification R4 is Equinox[5] which is integrated inside the Eclipse IDE to make easy development. It is actively-maintained and massively user-based. However, using the IDE for running and testing bundles on Equinox framework causes problems later when it is used in stand-alone. Moreover, Equinox drags some overweight into the system with features that are not needed. This usually doesn't cause a problem when running the framework on a server, but when Equinox is integrated inside an embedded device, this might matter.

Apache Felix The Apache Felix[6] is a lightweight implementation, easy to use in standalone, suitable for small device integration and implements the OSGi specification R4. It has almost 0% of CPU usage and 12MB of real memory usage and it is the most standard and generic for Distributed OSGi implementation.

Table 2.1 illustrates the characteristics of the OSGi implementations and the differences between them.

Table 2.1: Comparative table of OSGi implementations

	GUI based	Lightweight	R4 implementation	Compact device integration
Concierge	x	x		x
Knopflerfish	x		x	
Equinox		x	x	
Apache Felix		x	x	x

2.4 Services Interoperability

Two main communication mechanisms are possible for services and information exchange between bundles. They are substantially based on service and event exchange inside the same OSGi container, or even between different containers for a distributed OSGi communication.

[5]http://www.eclipse.org/equinox/

[6]http://felix.apache.org/

2.4.1 Service Exchange

The service communication among bundles in OSGi specification is established based on the service publish, find and bind SOA model.

Local Service Exchange

In local service exchange, service providers publish their services in the dynamic service registry while service requesters use the service registry to find services and bind to service providers. In OSGi, a service is defined by its service interface and the functionality it provides is implemented accordingly to this interface. Each bundle can register any number of services in the service registry using its interface name and its properties. Other bundles can query the service registry for services using their names and properties.

Distributed Service Exchange

The Distributed Open Service Gateway initiative (DOSGi) is introduced in the OSGi specification release 4.2 (R4). Distributed service exchange has solved a critical problem of the OSGi specification that is the communication between containers. This is quite relevant for pervasive communication between different devices hosting OSGi containers. DOSGi is implemented under the *Apache CXF project*[7] for *Apache Felix* and it allows remote services invocation and the dynamic service discovery mechanism. The principle is the same as for local service exchange with the introduction of the DOSGi bundle in the different distributed containers for managing service export and import between them. The open source server for highly reliable distributed coordination, *Apache Zookeeper*[8] can be used as a centralized service registry so that service requesters could lookup for one service in all the available containers as shown in Figure 2.4.

2.4.2 Event Exchange

Event communication among bundles in OSGi is based on the publish/subscribe paradigm, where event publisher bundle sends an event related to a specific topic while an event-subscriber bundle expresses interest in one or more topics and receives all the messages belonging to such topics.

[7] http://cxf.apache.org/distributed-osgi.html
[8] http://zookeeper.apache.org/

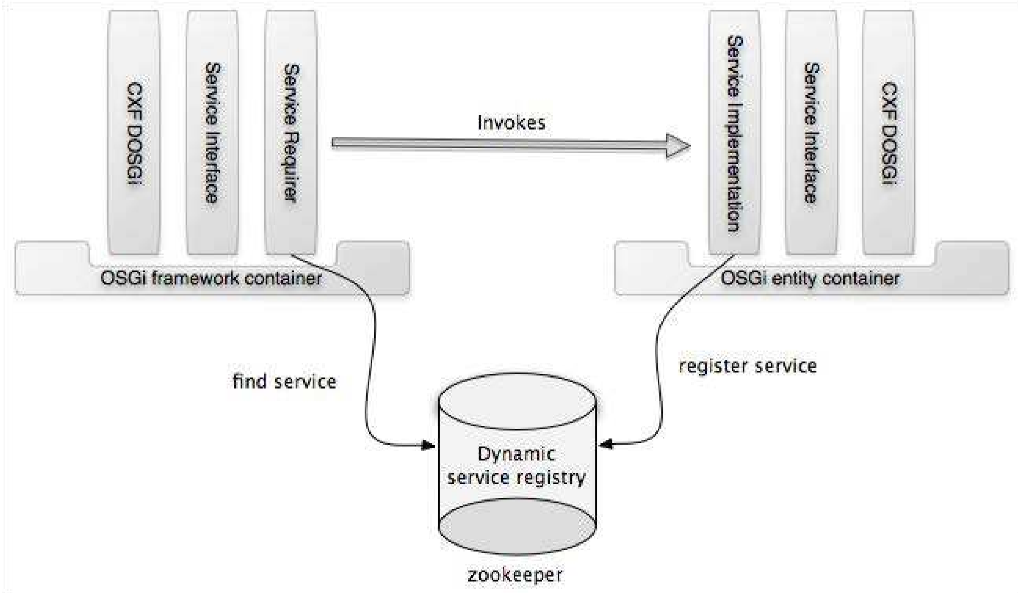


Figure 2.4: Distributed service invocation in OSGi

Local Event Exchange

Based on the *EventAdmin* service of OSGi, event publisher bundle will send its events using a specific topic to the *EventAdmin* service which will forward them to all event consumers bundles subscribed on the same topic.

Distributed Event Exchange

Using the same paradigm, it is possible to reproduce the event exchange mechanism in a distributed way between bundles hosted on different OSGi containers. This can be achieved using Java Messaging Service (JMS) or other messaging frameworks. Both ActiveMQ[9] and XMPP[10] protocols can be used to implement this mechanism.

2.4.3 The Device Profile for Web Services Communication Mechanism

The DPWS [60] is a web service standard. It is the successor of the Universal Plug and Play (UPnP) mechanism as in essence it allows entities in the environment to discover each other's presence on the network and specify a protocol for interacting with services offered by these entities [61]. In this way, entities will be able to communicate with each other or with other DPWS enabled applications using a unified and standardized protocol. DPWS

[9] http://activemq.apache.org/
[10] http://xmpp.org/

includes ... l event-
ing. Bo ... ds such
as WS- ... sing the
WS-Dis ... es stack
with th ... enable
discover ... entation
of this ... ks [62].

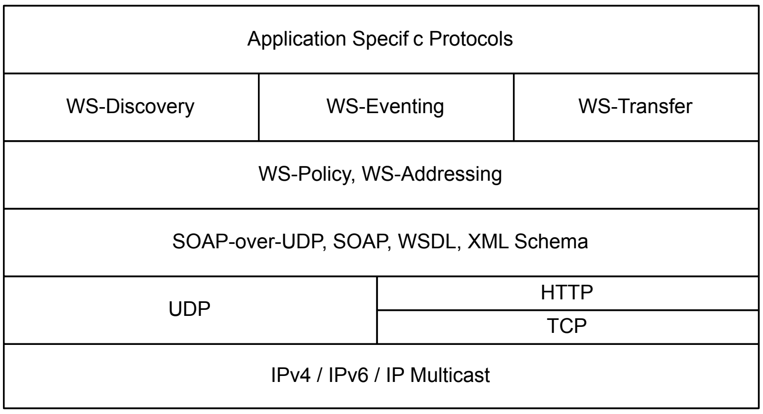


Figure 2.5: the Device Profile for Web Services Stack

A comparison between the three different mechanisms for distributed communication is summarized in Table 2.2

Table 2.2: Comparative table of OSGi distributed communication mechanisms

	Service Exchange (DOSGi)	**Event Exchange (ActiveMQ)**	**DPWS**
Service based	x		x
Event based		x	x
Entity description	Interfaces	XML schemas	WSDL
Standard			x
Security	Security layer		WS-Security

[11] RFP 86 DPWS Discovery Base Driver, OSGi Alliance

2.5 Knowledge Modeling & Reasoning

2.5.1 The Declarative Approach vs. the Imperative Approach

To implement the reasoning aspect of a service-oriented platform, one could opt for a classical imperative approach, which is very robust and requires only a short design phase. However, this brings deep constraints in term of re-usability in personalized environments and adaptability in dynamic environments. We consider that a declarative approach allows a more efficient separation between application logic and underlying models describing the peculiarities of the environment. Although this choice represents an important trade-off on the effort to be put at the design phase, it seems unavoidable to ensure the dynamism of the framework, essentially when targeting a large scale deployment. To adopt a declarative approach, several options are available. Strang and Linnhoff-Popien [63] identify and compare six types of context models: attribute-value pairs, schema-based models, graphic models, logic-based models, object-oriented models and semantic-based models. From this study it was deduced that the semantic-based models are the most complete and expressive ones. Hence they are the most suitable for modeling context for AAL. As a matter of fact, semantic web technologies provide a state-of-the-art modeling syntax with reasoning capacity and it is fitted by nature to highly-dynamic and open application domains. These technologies provide a level of abstraction common to all entities and bring down to each entity the possibility to understand other newly-discovered entities and to create bindings with the environment.

Semantic web technologies have been widely used in the domain of context understanding. The OSGi-based infrastructure for context-aware applications presented in [58] uses the notion of ontologies for context modeling and a semantic reasoning is performed to cope with the changing context. Ricquebourg et al. [64] also propose a semantic context model for smart home that represents the different persons, locations, appliances and sensors in the environment. Contextual inference rules are used for home automation. ELDeR ontology is proposed in [65] to represent and identify the possible risks in elders' homes. Other ontology-based context models have been designed for health monitoring and alert management for chronic patients care at home [66, 67]. Ontologies for context representation have also been expanded for user profile modeling [68].

2.5.2 Semantic Web Technologies

Semantic Modeling

A key requirement for modeling AAL environment and context is to give AT systems the ability to understand their situational conditions. In order to achieve this, contextual information should be adapted for machine processing and reasoning. The semantic web technologies are selected as the adequate solution to fulfill this requirement [63]. Even if Semantic Web standards and the supporting tools are originally designed for Web-based applications, they are well suited for many requirements of AAL environments. In fact, knowledge representation languages used in Semantic Web have rich expressive power that is adequate for modeling various types of contextual information such as information associated with people, sensors, devices, places, time, and space. Ontologies created through semantic modeling do not only have the advantage of enabling the reuse and sharing of common knowledge among several applications [69], but also of allowing the use of logic reasoning mechanisms to deduce high-level contextual information [70]. In context-aware applications, Semantic Web inference engines use specific logical rules to deduce implicit, higher-level context from explicit, lower-level context. Furthermore, Semantic Web query service can be leveraged to facilitate expressive context query, allowing applications to access context through the use of declarative queries.

This separation of the application logic and its underlying models is considered as a huge support to the SOA approach for the achievement of complete dynamic and adaptable AAL systems. Even though the service-oriented approach allows the representation of the different environment elements as services, they are still not integrated into the application logic. The semantic modeling ensures this by adding the semantic representation of these elements to the ontology which implies their implication into the reasoning mechanism through the Semantic Web inference engine. Moreover, the service-oriented approach only provides a mechanism where entities discover each other and start exchanging data. However, entities actually do not know about each other's bindings with the environment. E.g. where a sensor has been deployed? Who is carrying the interaction device? Etc... Being able to parse data received from a new unknown entity is not enough. A semantic description of this entity and its bindings is needed.

Resource Description Framework (RDF) [71] is a standard metadata data model for making semantic descriptions. Built based on XML which is used for syntax, RDF contains a clear set of rules for providing simple descriptive information at a semantic level. RDF Schema (RDFS) then provides a way to combine multiple RDF descriptions into a single vocabulary. RDF is based on a concrete formal model that uses directed graphs for representing the semantics of metadata. The core of every RDF expression consists in a (subject, predicate, object) triple. Every triple consists of: the subject (resource being described), predicate (named property), and object (the value of this property). Resources and predicates are represented by URIs. This abstract syntax of RDF is serialized using several alternative concrete syntaxes like RDF/XML [72], Notation3 (N3) and N-Triples [73].

Semantic Reasoning

Knowledge can be acquired through pure reasoning alone (rationalist approach) or via experiences as perceived through the senses and stored in the memory (empiricist approach) [74]. In AAL, a mix or a compromise between these two approaches can be considered where context is first perceived via the sensors and stored into ontologies. Reasoning is then employed to transform contextual data into meaningful information and infer new, implicit context information that is relevant for the AT. Semantic Web technologies are well fitted to this approach with the semantic representation of the knowledge and sensed information using ontologies. Once context information is represented, it can be reasoned about in a logical way using semantic reasoners.

Several available semantic reasoners can be used here. Among them: Jena, Pellet, RacerPro and EYE. Each of them has its proper characteristics that differentiate it from the others and promote its use in specific domains [75].

In the domain of AAL, reasoning is usually related to the context understanding. The context-aware reasoning can be categorized into four main perspectives [76]:

- The low-level perspective: it includes basic tasks such as data pre-processing, data fusion and context inference. It is usually performed by the sensors or the Middleware.
- The application-oriented perspective: where the application can use a wide variety of reasoning methods to process the context data.
- The context monitoring perspective: the main concern at this level is a correct and efficient update of the knowledge base as the context changes.

- The model monitoring perspective: the main task is to continuously evaluate and update learned context, taking into account user feedback.

2.6 Uncertainty Handling

One very important requirement for reasoning with context information is to deal with uncertainty. Context reasoning for AmI is very complex due to dynamic, imprecise, imperfect and ambiguous nature of context data which may derive from technological failures, sensing imperfection or incomplete knowledge. In fact, uncertainty is classified into two types [77]:

- Aleatory uncertainty: results from the fact that the system behaves in random ways. It is also known as stochastic uncertainty or type A uncertainty.

- Epistemic uncertainty: results from the lack of context knowledge. It is also known as subjective uncertainty or type B uncertainty.

To deal with these problems, a notion of uncertainty representation and reasoning needs to be introduced in AAL solutions. Different research works have been achieved in this perspective. In the domain of activity recognition and prediction, fuzzy clustering methods have been used to perform activity learning in order to derive patterns of scenarios. Then comparison between these patterns and the actual observation will allow predicting the end-user current activity [78, 79]. This approach implies a long and time-consuming learning process to create patterns representing all the patient activities. It is mainly based on the *Epistemic uncertainty.* Other research works have combined fuzzy logic and the Q-learning to deal with the uncertainty, which provide more accurate results to deal with prediction issues [80]. However, this approach also relies on learning of human habits which is not efficient with end-users suffering from unstable lifestyle, as it is the case for elderlies with mild dementia. A comparison between the dynamic Bayesian network and the Dempster-Shafer theory approaches for activities recognition under uncertainty is elaborated in [81]. Uncertainty is also used in systems for location tracking [82] based on radio frequency waves generated by PDAs, using a back propagation neural network. The uncertainty in this case is mostly related to the *Aleatory uncertainty.*

2.7 Discussion

In this chapter, the main tools and approaches supporting the dynamism and the adaptability of AAL solutions in heterogeneous environments have been presented. An opening on uncertainty handling concluded the chapter. In this section, I will explain and argue our choices of the approaches we have used during this thesis.

First, we have chosen to use the modular approach through SOA to ease the extension of AAL framework with the integration of new assistive services and new sensing and interaction devices in the assisted person environment. As SOA is only an abstract approach, two different implementations are possible, the web service programming and the OSGi. Although the use of web service is beneficial in the case of wide environment and communication between very far entities, for small and medium environments, this approach has many limitations, mainly the dynamic adaptation to the framework changes. The use of OSGi appears as a suitable substitute. In fact, the Open Service Gateway initiative is developed as a home gateway that supports services life cycle and allows integrating or removing services at runtime. It provides several useful Application Programming Interfaces (APIs) to manage systems composed of services and has been used in numerous projects for smart home development [55, 56, 59]. It also allows the communication with web services which keeps the possibility to use some web-based services. Some criteria have been defined for the choice of the appropriate OSGi implementation for AAL development, mainly the possibility of integration in small and tiny devices, the framework weight and the implementation of the R4 of the OSGi specification. R4 specifies especially the multi-containers communication, which is important in multi-devices information exchange. As it is showcased earlier in this chapter in Table 2.1, the Apache Felix is the most convenient and appropriate implementation according to our fixed needs. Therefore it has been used for our implementation. For the services interoperability part, different communication mechanisms, based on service and event exchange, have been presented in this chapter and have been used in our implementation.

Second, we have chosen to use the semantic approach to represent and reason about knowledge. This allows integrating the declarative approach by separating the application logic from the underlying models which is crucial for a dynamic and adaptable framework. The use of semantic web to implement this aspect has four main purposes in our use-case of AAL: 1. the modeling of assistance in smart spaces, 2. the integration of all discovered entities in the system process, 3. the collaboration between different modules of the system based on a shared model and lexical and 4. the reasoning to create the system's intelligence, based on the three previous points. The last point is performed using semantic matching between the knowledge about users' context, derived from sensor events and formalized into an ontology, and respectively services' and devices' semantic profiles. To implement the reasoning aspect, I have used Eulersharp [12] (EYE) based on the comparison performed in [75]. Euler has two advantages applicable to any use-case: its scalability, due to its optimized implementation based on YAP-Prolog, and its human readable formalization language using N3. However, Euler is a naive (memoryless) reasoner, which is crucial from our perspective but might be counter-productive in many applications. Here lies the main trade-off in our choice. It is notably among the fastest reasoners we have found and is also the most lightweight of the reasoners we have selected and compared.

Finally, a need for uncertainty handling emerges from the state of the art on dynamic assistive service framework and semantic reasoning. In fact, semantic representation and reasoning does not concede imprecise information and deal with information as an absolute truth. In order to cope with sensors' imprecise data and missing or partial context information, we need to provide an approach for the measure of uncertainty and Quality of Information (QoI) and enhance the semantic representation and reasoning to take these measures into account. Different formalisms can be used to support the semantic reasoning under uncertainty.

Figure 2.6 represents the main aspects I have considered in my approach for handling the dynamism and the emerging need of uncertainty handling in AAL environments.

[12] http://eulersharp.sourceforge.net/

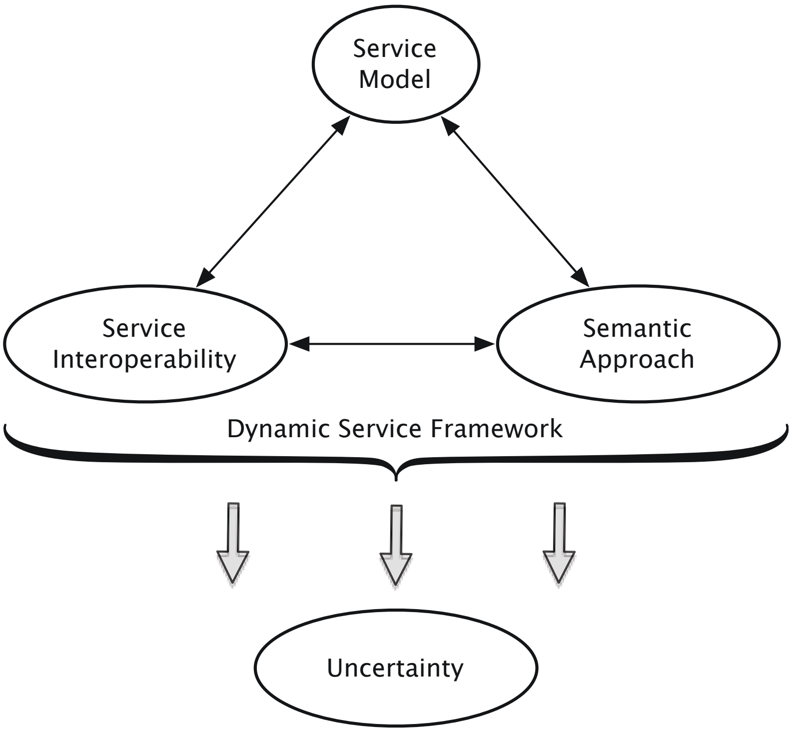


Figure 2.6: Different aspects for handling dynamism in AAL solution

3

Research Approach & System Design

3.1 Introduction

Based on the approaches and tools we have identified in the previous chapter, I have developed a dynamic and extendible ambient assistive living framework called Ubiquitous Service Management ARchiTecture (UbiSMART). The first realized version was based on the distributed event communication mechanism with manual integration of new assistive services, sensors, and devices. The framework has been upgraded later using our semantic Plug&Play approach to allow, at runtime, the dynamic integration of assistive services and their related entities.

In this chapter, I advocate our vision for the development of the UbiSMART framework, then I detail the evolution of the proposed framework from a static to a dynamic aspect. I conclude the chapter by the validation of the semantic Plug&Play approach I have proposed.

3.2 Vision of Ambient Assistive Living Systems

The basic idea of AAL systems is their ability to understand the real world and react accordingly by providing relevant information and/or services to the end-user. An AAL system is therefore composed of different sensors – e.g. pressure sensors, proximity sensors, vibration sensors or motion sensors – deployed in the patient environment. It holds also different devices of interaction – e.g. speakers, tablet PCs, smart-phones, or nursing consoles –. Finally, a software platform drives the context awareness and the reasoning for the provision of appropriate assistive services for the patients.

Sensors are used to monitor the patient and acquire low level context information while devices are used to provide assistive services for the patient. The software platform embeds a reasoning engine in charge of taking decision of the assistive service that should be provided to the patient and of the appropriate device of interaction that should be used to provide this service. This decision is based on low level context information received from the sensors and high level inferred context.

An AAL system consists therefore of three main components as shown in Figure 3.1:

- Micro-context reasoner: the micro-context reasoner is in charge of receiving raw data from the sensors and generating micro-context information such as temperature, distance, orientation, etc. micro-context information is elementary and related to sensors states.

- Macro-context reasoner: the macro-context reasoner composes micro-context information to infer higher level context and human behavior (location, activities, emotion, etc.)

- Service and device reasoner: the service and device reasoner uses macro context information in order to infer meaningful information and needed services by the patient, then decides on the device to use to provide the service.

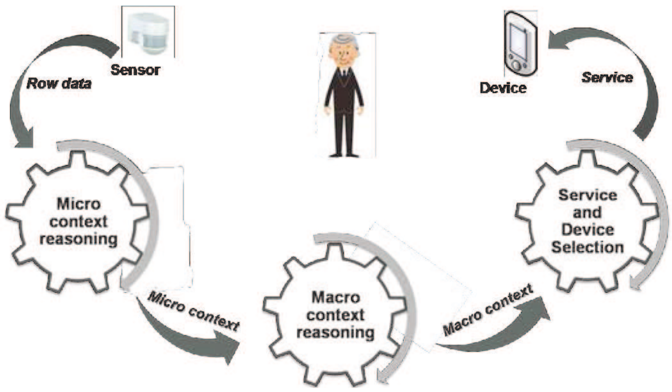


Figure 3.1: Principle components of the AAL system

These different components are connected to a centralized semantic model that describes the environment. The micro-context component feeds the model with new sensing devices through the registration of new detected sensors in the environment. It also adds information about sensors micro-context when it starts receiving sensors events. The macro-context component uses all the collected micro-context information in the semantic model to infer a higher-level context, and then updates the semantic model with the inferred context. On the service and device part, assistive services and interaction devices register themselves to the semantic model in order to take them into consideration to provide assistance to the end-user. Inference rules are applied to the semantic model in order to select the appropriate assistive service and the adequate interaction device, based on the current context of the end-user. Figure 3.2 represents an overview of the UbiSMART software framework.

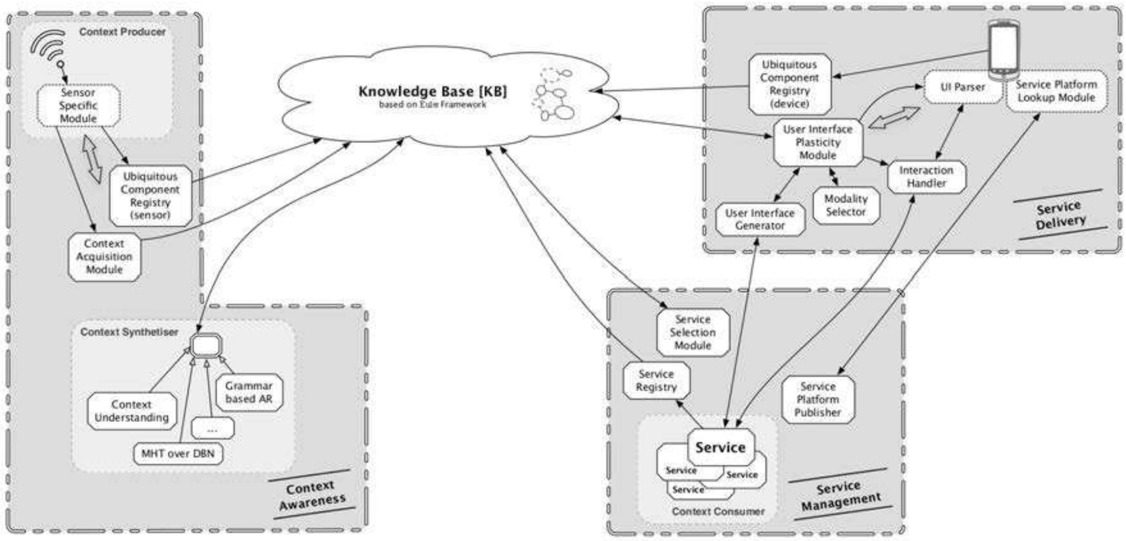


Figure 3.2: Overview of the UbiSMART software framework

3.3 The UbiSMART Framework

The different components of the UbiSMART framework are decomposed into various modules communicating with each other and running on different OSGi containers. This modular approach leads to develop a framework that covers sensors and devices deployed in different locations of the end-user environment. The different acts of assistance that should be provided are represented as assistive services which can be integrated or removed from the framework at runtime with smooth reconfigurations. Assistive services are classified into information services (reminders, alerts), acting services (remote control services), communication services (emergency call) and leisure services (cognitive games). The different sensing technologies and interaction devices are also represented as services (careful: not assistive services) in the framework. These services can provide context information or display an assistive service.

Assistive services, sensors' and devices' services are packaged into bundles. Leveraging the dynamic class-loading property in OSGi, these bundles can be started, stopped or updated independently at runtime without affecting other components or without restarting the entire framework. The dynamic service registry of OSGi allows bundles to register, discover and request for services at runtime. If a new act of assistance needs to be provided for the end-user, we just need to install the corresponding assistive service and bind the sensors and interaction devices related to it.

As discussed in the previous chapter, the framework should also be able to communicate with entities presented as web services. Different sensors and devices could be present in several environments as web services, given their WSDL files, these entities are integrated into the UbiSMART framework through the generation of specific client bundles installed on the OSGi containers. This was achieved using *Apache CXF*[1]. Client bundles consume the entities web services to get the sensors' events or to provide an assistive service. Figure 3.3 shows the OSGi representation as bundles of the different entities discovered in the environment.

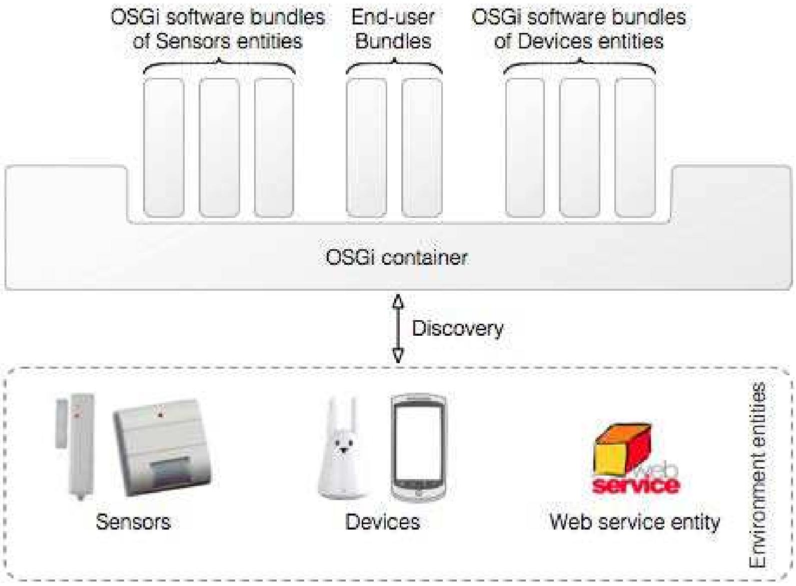


Figure 3.3: Entities representation on OSGi framework

3.3.1 Proposed UbiSMART Architecture

The UbiSMART framework is represented through the spider-gram in Figure 3.4. It is a generic framework able to communicate with different sensors and devices using multiple communication protocols. It uses different communication mechanisms between the different OSGi containers deployed in the environment, integrates new assistive services at runtime and manages the semantic reasoning for assistive services provision. Next, I define the different modules of our framework.

"X" Gateway The framework should be able to communicate with different sensors using heterogeneous sensors' communication protocols. Therefore, the "X" Gateway module encompasses all the modules managing the communication with these protocols. If

[1] http://cxf.apache.org/docs/developing-a-consumer.html

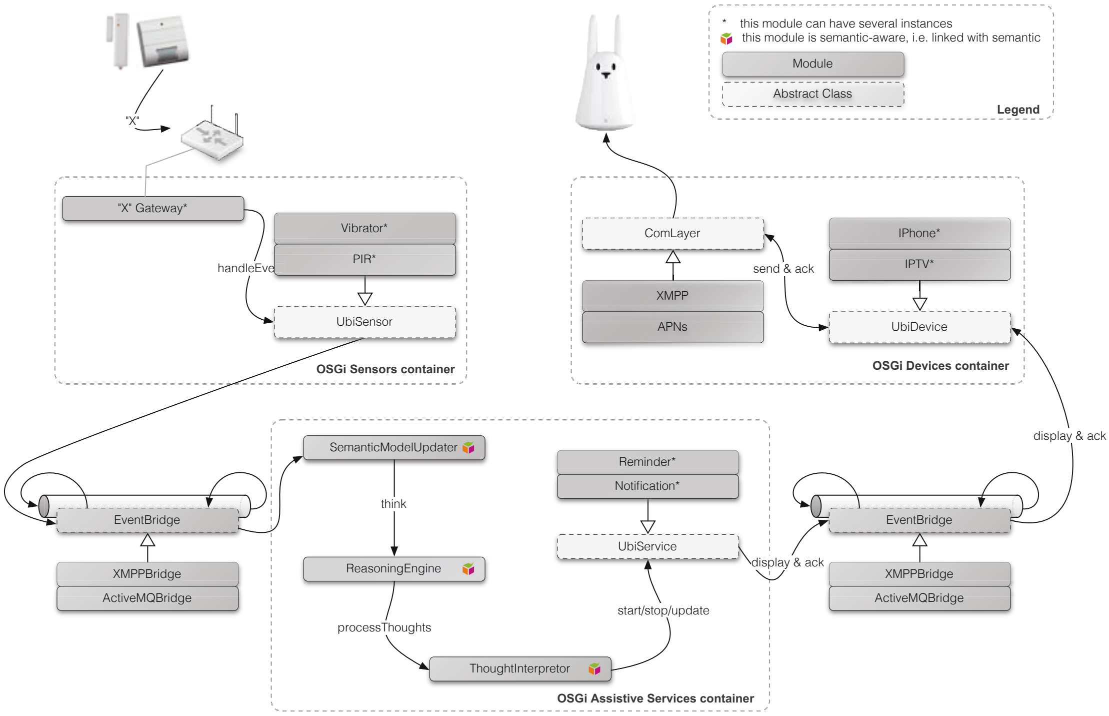


Figure 3.4: Ubismart framework architecture

we need to integrate new sensors with a different communication protocol, we just need to integrate their relatively developed gateway in our UbiSMART framework and these sensors' events become ready to be processed. Thus, different gateways could operate at the same time on the framework such as *"Zigbee Gateway"* or *"ANT Gateway"*.

UbiSensor This is an abstraction of the different modules representing the deployed sensors in the OSGi sensors container such as Passive Infra-Red (PIR) or vibrator sensors. Each module representing a sensor is in charge of receiving this sensor's events and making decision the way how they should be processed depending on the type of the sensor and the information that it should provide. In fact, sensors' raw data need to be processed to provide high-level information before it is transferred to the reasoning engine.

Event Bridge It is an abstraction of the different mechanisms of communication possible between OSGi containers (*XMMP Bridge, ActiveMQ Bridge ...*). This gives the flexibility to change the communication mechanism with smooth reconfiguration.

Semantic Model Updater This module is in charge of updating the semantic model with micro-context information received from the sensors OSGi containers. This allows our reasoning engine to infer macro-context information and to make decisions. In fact, each time new micro-context information is received, the changes in the semantic representation may lead to infer new macro-context information or to make new decisions.

Reasoning Engine This module is responsible for taking decisions based on defined first order logic rules and the evolution of the semantic representation of the environment once new sensors' events are received.

Thought Interpreter It analyses the output of the *Reasoning Engine* module, updates the semantic representation with new inferred information and transfers the decision on the assistive service and interaction device to use.

UbiService It is an abstraction of the different assistive services that can be provided such as *Reminder* or *Notification*. These services can be started, stopped or updated at runtime. Selected assistive service receives the identification of the device on which it will be provided. It invokes the device to display the assistive service through the *Event Bridge* module.

UbiDevice It is the abstraction of the interaction devices bundles such as *iPhone* or *IPTV*. Once the interaction device bundle is invoked, it sends the information to the interaction device through the *ComLayer* module.

3.3.2 Architecture Based on Distributed Event Communication

In order to ensure the communication between the OSGi sensors' container, the OSGi assistive services' container and the OSGi devices' container, I have used the distributed event communication with the publish/subscribe paradigm using ActiveMQ. Figure 3.5 shows our proposed architecture based on event communication.

ActiveMQ manages the event communication between the framework and the environment entities. Once a sensor in the environment sends a new event, the module representing this sensor on the OSGi sensors' container receives the event, makes processing on the raw data and sends the micro-context event on a specific topic (*sensor*) on which the OSGi assistive services' container is subscribed and is waiting for events. Furthermore, each interaction device is subscribed to its own topic (such as *console* for a nursing console), and is listening for events from the framework on the same topic to render the

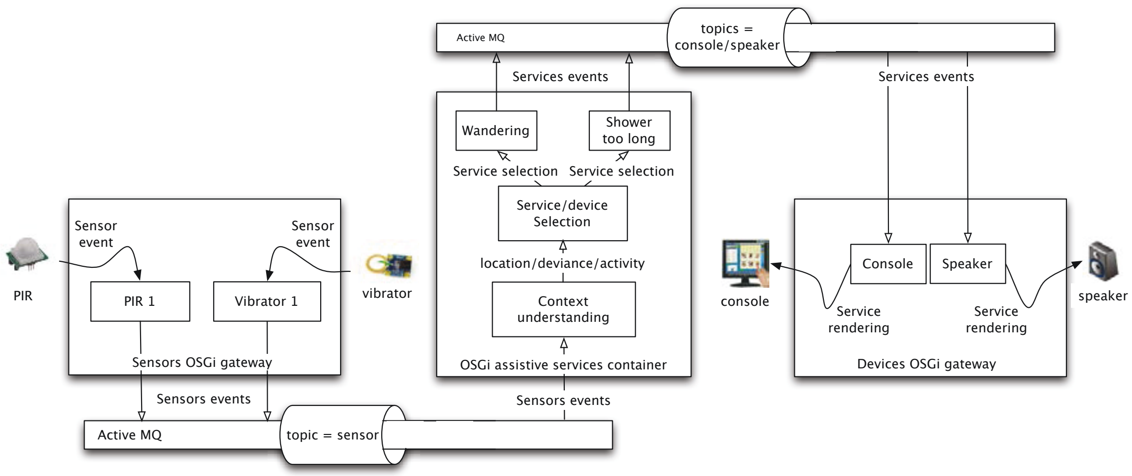


Figure 3.5: Architecture based on the distributed event communication

assistive services. The event syntax used for communication is based on the XML format. Two XML schemes have been identified as shown in Figure 3.6.

```
<sensor>
    <sensorType>PIR</sensorType>
    <sensorID>1</sensorID>
    <sensorLocation>washroom</sensorLocation>
    <event>Firing</event>
</sensor>
```

(a) Sensors events XML schema

```
<service>
    <patient>patient_1</patient>
    <location>washroom</location>
    <activity>showertoolong</activity>
    <type>Error</type>
</service>
```

(b) Services events XML schema

Figure 3.6: XML schemes for event communication

The event communication approach gives flexibility as there are transparency between the environment entities and the framework. However, each time a new sensor or interaction device needs to be integrated, new topics need to be added and the framework needs to be recoded to take into account the newly integrated instances. In addition, the defined XML schemes are not standardized which may cause some communication problems.

3.4 Handling Dynamism in the UbiSMART framework

New modules have been integrated to the UbiSMART architecture to handle the dynamism in the framework so that new assistive services, sensors and interaction devices can be integrated in the framework at runtime. In addition, I have used the DPWS communication mechanism which is a standard communication mechanism allowing to represent new devices' profiles (sensors or interaction devices). It also provides a discovery mechanism completely transparent. The combination of the DPWS mechanism with the automatic integration of newly-discovered entities' profiles into the semantic model produces the needed dynamic aspect. Figure 3.7 represents the adapted UbiSMART framework.

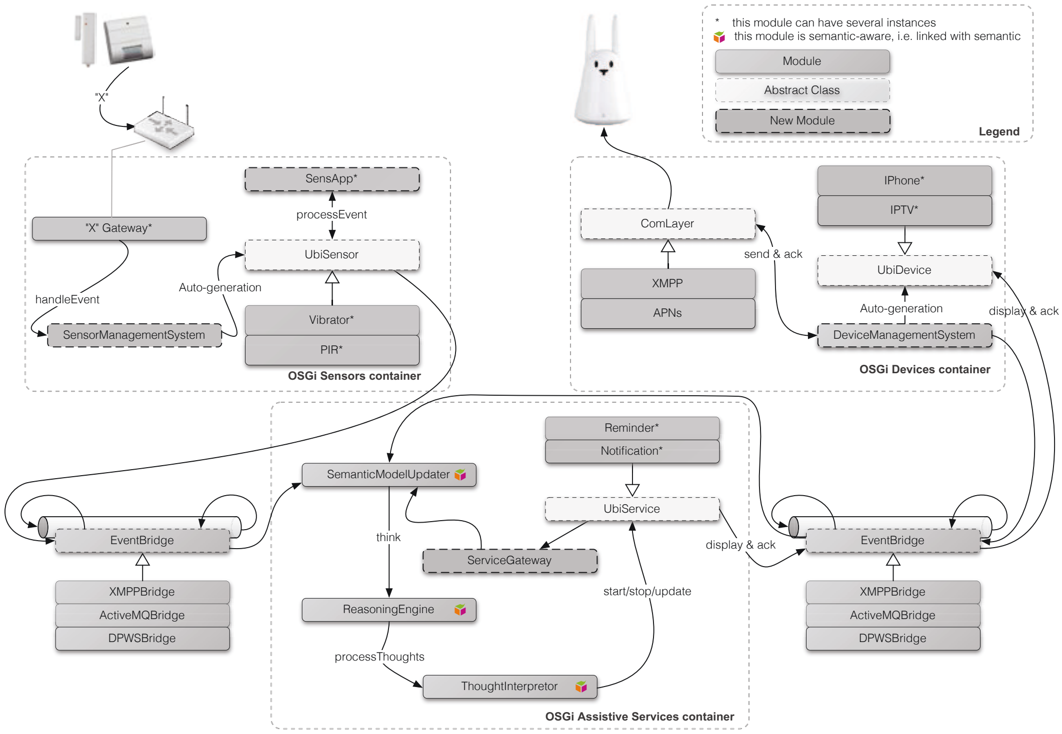


Figure 3.7: Updated UbiSMART architecture handling dynamism (to be compared with Figure 3.4)

New integrated modules mainly allow the semantic integration of newly-integrated entities to the semantic model of the framework:

Sensor Management System It is in charge of the auto-generation of modules representing new detected sensors in the environment. It is also responsible for defining the possible use of these new sensors. Depending on the intended use of the sensors, the processing needed for events received from these sensors may change. The *Sensor Management System* module routes the received events to the adequate *SensApp* module for processing.

SensApp This module represents the different modules responsible for the micro-context processing of sensors' events and all other needed events-related processing before transferring the generated micro-context events to the reasoning engine.

Service Gateway This module is responsible for detecting the integration of new assistive services in the framework and their semantic registration through the *Semantic Model Updater* module.

Device Management System This module has a similar behavior to the Sensor Management System (SMS) module. It auto-generates modules representing newly-detected interaction devices in the environment. It also adds bindings of these devices to entities in the environment.

Semantic Model Updater The *Semantic Model Updater* module was modified to include new functionalities, mainly the integration of the semantic representation of newly detected sensors, devices and assistive services with their binding in the environment.

3.4.1 Architecture Based on Device Profile for Web Services

The use of the DPWS communication mechanism provides a base for the dynamic aspect of the system through a Plug&Play mechanism allowing the representation as a service of any sensor or device discovered in the environment at runtime. This mechanism is detailed in Figure 3.8.

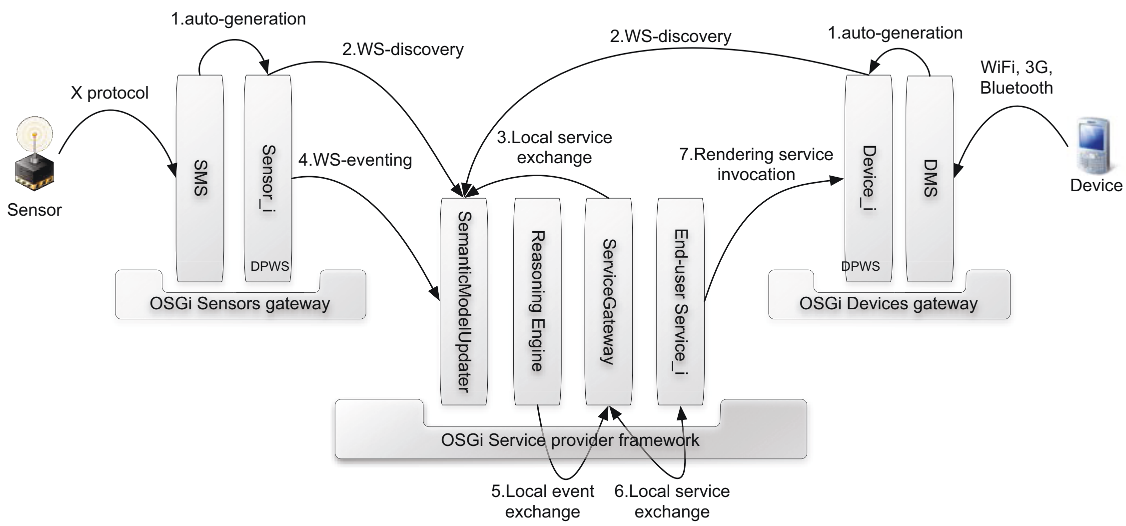


Figure 3.8: Detailed architecture of the semantic Plug&Play mechanism

The SMS bundle on the OSGi sensor gateway automatically generates a DPWS-based bundle representing each new sensor detected in the environment (1 in Figure 3.8). The generated bundle contains a description of the sensor and its functionality – type, ID, events, etc –. The type and ID information are deduced from the sensor's packets received through the communication protocol. A similar process is also implemented on the OSGi device gateway where the Device Management System (DMS) communicates with devices through WiFi, 3G or Bluetooth and automatically generates DPWS based bundles describing these devices – type, id, location, rendering capability (audio, video, picture, text, light, etc.)– each time a new device is discovered (1 in Figure 3.8). SMS and DMS generate these bundles and start them when new entities are discovered, then stop and remove them when the corresponding entities are removed from the environment. The DPWS communication protocol allows the discovery of the new entities by the service provider framework (2 in Figure 3.8) and sending events when sensors change their status (3 in Figure 3.8) or when an assistive service needs to be rendered on a device (7 in Figure 3.8). Appendix C provides more technical information on this mechanism. The mechanical Plug&Play mechanism described up to this point, allows the integration and representation of sensors and devices as services into the framework. However, these entities are still not integrated into the reasoning process as no information has been provided about their bindings to the environment (who is using the device? where are the sensors actually deployed? etc.). Thus it is not possible to use them in the selection of the end-user service and the interaction device, which induces the need of a semantic Plug&Play mechanism.

This approach provides a complete flexible and dynamic framework with the ability to add or remove sensors and devices at runtime without the need to recode the framework. It also allows attaching and configuring new assistive services with their related sensors and interaction devices in total transparency between the framework and the different entities.

3.4.2 The Semantic Plug&Play Mechanism

Even though the mechanical Plug&Play mechanism discussed earlier allows the integration and representation of assistive services, sensors and devices as services into the framework, they are still not integrated in the reasoning process. Thus it is not possible to use them for the detection of new context information and the selection of the appropriate end-user assistive service and interaction device. We call this a mechanical Plug&Play. I have introduced the notion of semantic Plug&Play to solve this problem. To integrate new assistive services, sensors and devices in the process, the *Semantic Model Updater* module detects new entities when it receives their descriptions and updates its representation of the environment with the semantic representation of these entities, extracted from their discovery events. It also detects the installation of new assistive services and adds them to the semantic model.

Assistive Service Semantic Plug&Play

Once an assistive service is defined with its related sensors and interaction devices, it should be integrated into the reasoning process of the framework. We consider that an assistive service is related to an abnormal context, and it intends to incite end-user to solve this situation in order to achieve a normal context. Therefore, when a new assistive service is integrated in the framework, it provides its description with the related contexts to the semantic model (4 in Figure 3.8). A specific rule is integrated into the reasoning engine in order to trigger assistive services once their related abnormal contexts are detected.

Sensors/Devices Semantic Plug&Play

Newly-generated bundles, representing sensors and devices, use the DPWS discovery protocol, WS-discovery (2 in Figure 3.8) to advertise themselves and send their descriptions to the *Semantic Model Updater* module integrated in the OSGi service framework. A simple configuration tool on the OSGi service provider framework, presented in Figure 3.9, is in charge of adding bindings between the discovered entities and other objects in the environment in order to configure their use in assistive services provision.

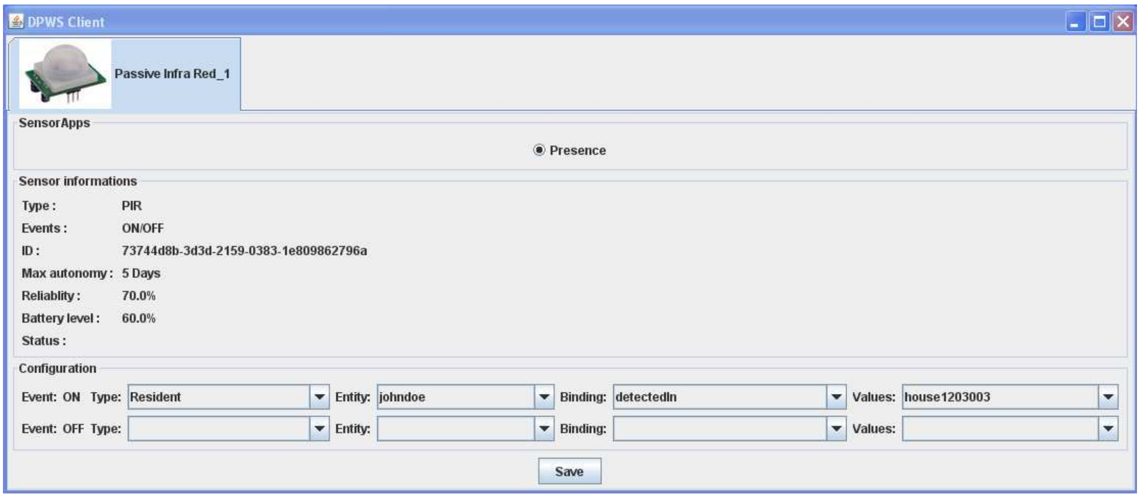


Figure 3.9: Configuration tool for semantic Plug&Play binding

The *Semantic Model Updater* module is in charge of adding all these entities' semantic presentation to the semantic model in order to include them into the assistive service and interaction device selection process by the *Reasoning Engine* module. It instantiates the semantic representation model of the detected entities using their semantic description. Next are the semantic model of sensors and the instantiated description of a detected PIR sensor using the N3 syntax:

```
### MODEL ###
  aal:Sensor a rdfs:Class.
  aal:SensorType a rdfs:Class.
  aal:SensorState a rdfs:Class.
  aal:hasId a owl:DatatypeProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang xsd:string.
  aal:hasType a rdf:ObjectProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang aal:SensorType.
  aal:hasPossibleState a rdf:ObjectProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang aal:SensorState.
  aal:hasMaxAutonomy a owl:DatatypeProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang xsd:int.
  aal:hasReliability a owl:DatatypeProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang xsd:double.
  aal:hasBatteryLevel a owl:DatatypeProperty;
      rdfs:domain aal:Sensor;
      rdfs:rang xsd:double.
### DATA ###
  hom:PIR a aal:SensorType.
  hom:pir_73744d8b_ON a aal:SensorState.
  hom:pir_73744d8b_OFF a aal:SensorState.
  hom:pir_73744d8b a aal:Sensor;
      aal:hasType hom:PIR;
      aal:hasID 73744d8b-3d3d-2159-0383-1e809862796a;
      aal:hasPossibleState hom:pir_73744d8b_ON;
```

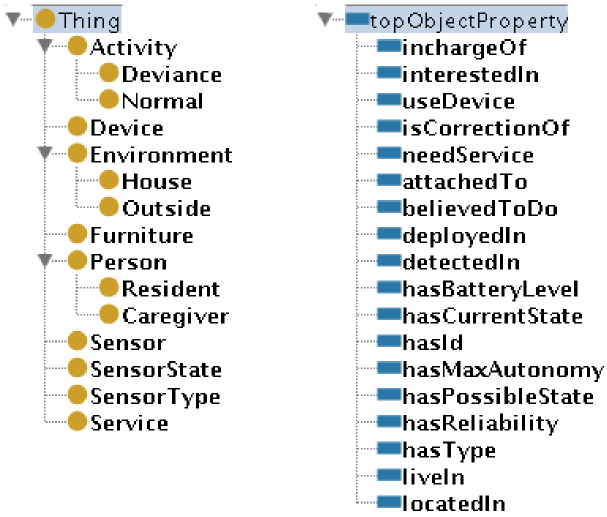
```
aal:hasPossibleState hom:pir_73744d8b_OFF;
aal:hasMaxAutonomy 5;
aal:hasReliability 70;
aal:hasBetteryLevel 60.
```

Once a new assistive service and its related sensors and interaction devices are integrated in the environment, the reasoning engine starts receiving events from the sensors (3 in Figure 3.8), assistive service and interaction device selection is performed, and the assistive service is rendered on the selected device (5, 6 and 7 in Figure 3.8).

3.4.3 Semantic Context Modeling and Reasoning

Our system is composed of several sensors deployed in the environment to perform some monitoring of the residents' behavior, and a set of interaction devices used to render the assistive services. A reasoning process needs to be integrated in order to perform the context understanding based on sensors' events and select the adequate assistive services for the residents when needed. This reasoning process also needs to select the suitable devices of interaction for the provision of the selected services. The selection of suitable assistive services and interaction modalities is performed by a rule-based semantic reasoner realizing a semantic matching between the knowledge about users' context derived from sensors' events, and respectively services' and devices' semantic profiles acquired from a global semantic model (ontology). We provide a semantic model that represents the knowledge about entities in the environment (including users and their activities, locations, assistive services, sensors and devices) and the relations between them. Figure 3.10a represents different classes we have in this model while Figure 3.10b shows the possible properties between them. Appendix D gives more details about the semantic model and reasoning.

In Figure 3.10a, the class `Environment` represents the different scenes of the patient's environment, such as bedroom, shower-room or toilet. `Furniture` represents the different objects of the environment. `Device` and `Sensor` classes contain all the sensors and interaction devices detected and connected to the framework. The `SensorType` class represents the different possible types of sensors while `SensorState` class represents the different status that sensors might have. The `Resident` and `Caregiver` are represented under the class `Person`. In this work, we are tracking specific patients' activities divided into abnormal behaviors and normal activities supposed to solve them. Abnormal behav-



(a) Classes (b) Properties

Figure 3.10: Classes and properties of our ontological model

iors are classified in the `Deviance` class while solving activities are categorized under the `Normal` class. The provided assistive services are under the `Service` class.

In Figure 3.10b, the `hasPossibleState` relation represents all the possible states that a sensor might have while `hasCurrentState` relation indicates the current status of a sensor. `hasId`, `hasType`, `hasbatteryLevel`, `hasReliability` and `hasMaxAutonomy` are intrinsic properties of devices and sensors. The relation `liveIn` specifies the environment in which a resident is living. The `detectedIn` relation marks the location of a patient while the `deployedIn` relation represents the location in which a sensor or a device is deployed. The relation `attachedTo` indicates the furniture to which a sensor or a device is attached. If a resident is performing a new activity, the relation `believedToDo` is created between him and this activity. The `needService` relation indicates the correspondence between a deviance activity and a service, while the `isCorrectionOf` relation shows that a Normal activity is the correction of a Deviance activity. The relation `interestedIn` is created once a resident is in need of an assistive service. If the resident needs assistance from a caregiver, the relation `isInchargeOf` is created between a caregiver and this resident. If an assistive service is to be rendered on a specific device for the patient, the `useDevice` relation is created between them.

Most relations between objects of the classes (so called individuals) do not exist in the initial state of the ontology as the system does not have any information about the end-users and their context. As an example, lets consider in Figure 3.11 an initial ontology of a room hosting two patients with mild dementia in a nursing home. Only the assistive service `wanderingAlert` allowing to detect a patient wandering during the night is deployed in the framework with its corresponding sensors (`pirBedroom`, `pressureBed1` and `pressureBed2`) and interaction devices (`iPhone` and `android` for the caregivers, and bluetooth `speaker1` for the patient). At this moment, most of the relations between the users and other individuals of the ontology do not exist as the reasoning engine still does not receive sensors' events.

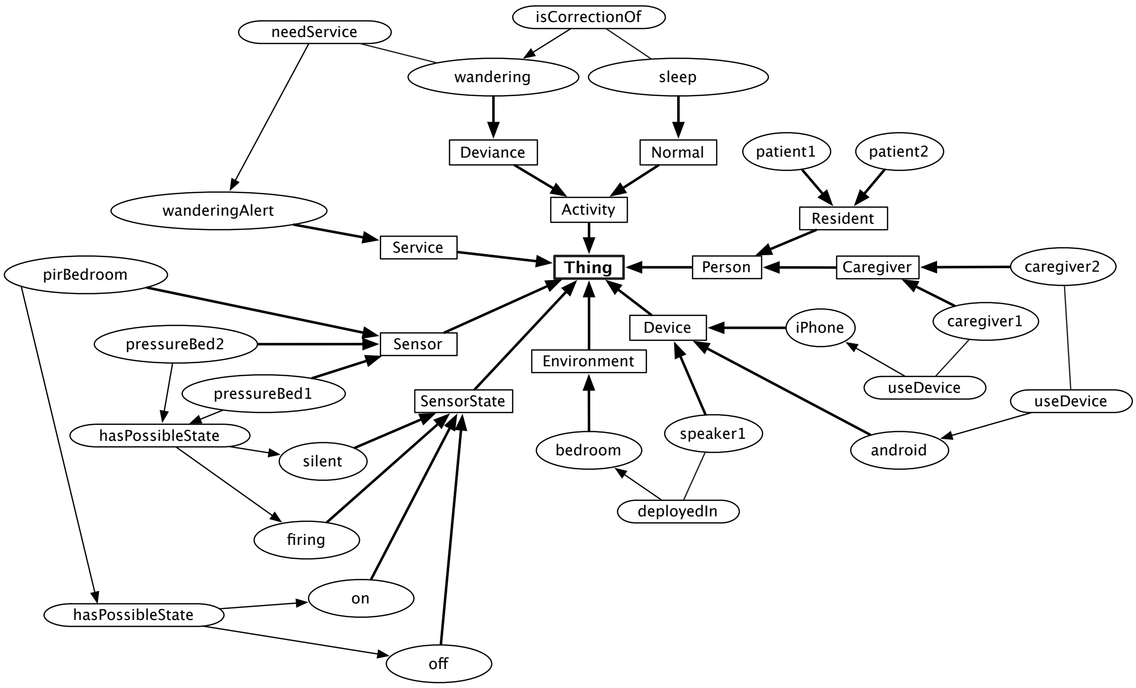


Figure 3.11: Initial Ontological Model

When the system starts receiving events from sensors, the ontology is automatically updated and a set of inference rules is applied to infer the context of the user, select the adequate assistive service depending on this context, and select the appropriate device of interaction. Used inference rules are presented below:

<u>*Service selection inference rule*</u>

$\forall$ *User u; Deviance dv; Service s*

(u, believedToDo, dv) $\wedge$ *(dv, needService, s)* $\wedge$ *log:notIncludes (u interestedIn s)*

$\Rightarrow$ *(u, interestedIn, s)*

<u>*Device selection inference rule*</u>

$\forall$ *User u; Location l; Device d*

$$(u,\ detectedIn,\ l) \wedge (d,\ deployedIn,\ l)$$
$$\Rightarrow (u,\ useDevice,\ d)$$

Figure 3.12 results from a context change where the patient needs the assistive service `wanderingAlert`. During the night, `patient2` is sleeping while `patient1` starts wandering. The relations *(pressureBed1, hasCurrentStatus, silent)* and *(pirBedroom, hasCurrentStatus, on)* are created. The reasoning engine infers that both the two patients are located in the bedroom, and that `patient1` is wandering while `patient2` is sleeping. The respective relations are created. First order logic rules infer that the `patient1` is interested in the `wanderingAlert` service. As this service is triggered during the night and in order not to disturb other patients, the reminder is not sent on the speaker, and the relation *(caregiver1, inChargeOf, patient1)* is created. A notification is therefore sent to the `caregiver1` on his/her `iPhone`. The ontology continues its evolution and it is updated each time the context of the patient changes.

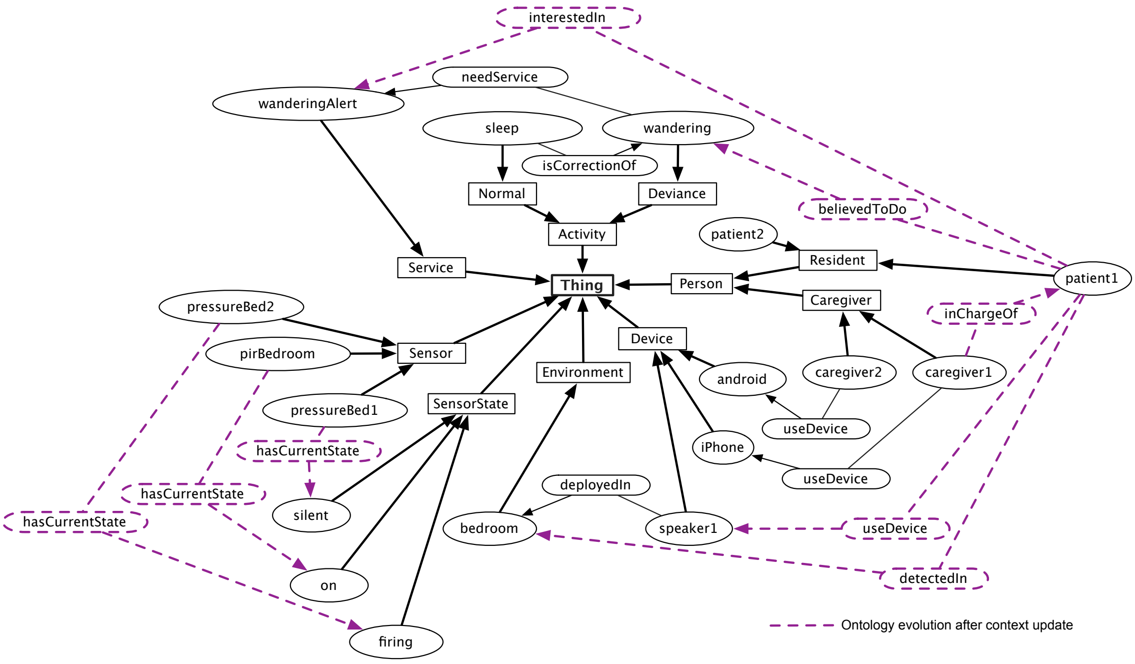


Figure 3.12: Ontology evolution after a context update

After the semantic Plug&Play of an assistive service and its related sensors and interaction devices, the semantic representation of the environment is alimented with these new entities. Figure 3.13 shows an example of a new assistive service (`Showertoolong`) integration into the framework with its related sensors (`pirShower` and `vibratorShower`) and interaction device (`speaker2`). New relations are also created based on sensors and devices description and configurations from the configuration tool.

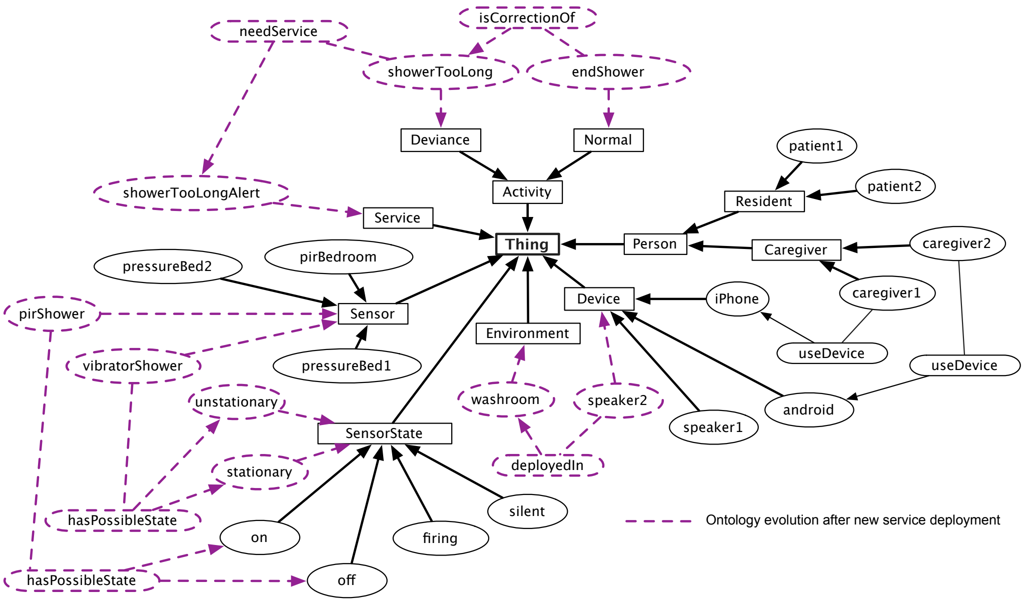


Figure 3.13: Ontology evolution after new service deployment

3.5 Validation of the Semantic Plug&Play Mechanism

The semantic Plug&Play mechanism composed of the OSGi Sensors gateway running into a tiny fanless Debian machine and the OSGi assistive services container running on a nursing console machine was validated. The assistive service provider framework includes Euler reasoning engine with a configuration tool showing the detected sensors, their description, and the potential events received.

The system was tested with low-cost sensors, used in our deployment, such as pressure sensors, accelerometers, ultrasound and PIR. These sensors are using ZigBee communication on a wireless-sensor network based on Crossbow's IRIS mote platform. When one of these sensors is activated in the environment, it is detected by the sensors gateway. This is reflected in the configuration tool which displays the detected sensor and its specifications. It is then possible to configure the use of the sensor in the framework functionality from the configuration tool. Euler will use this configuration to infer users' context in the environment. Figure 3.14 shows the scenario of integrating an ultrasound sensor in the framework.

The framework based on the semantic Plug&Play mechanism was tested with low cost sensors, used in our deployment, such as vibrator sensors, proximity sensors and motion

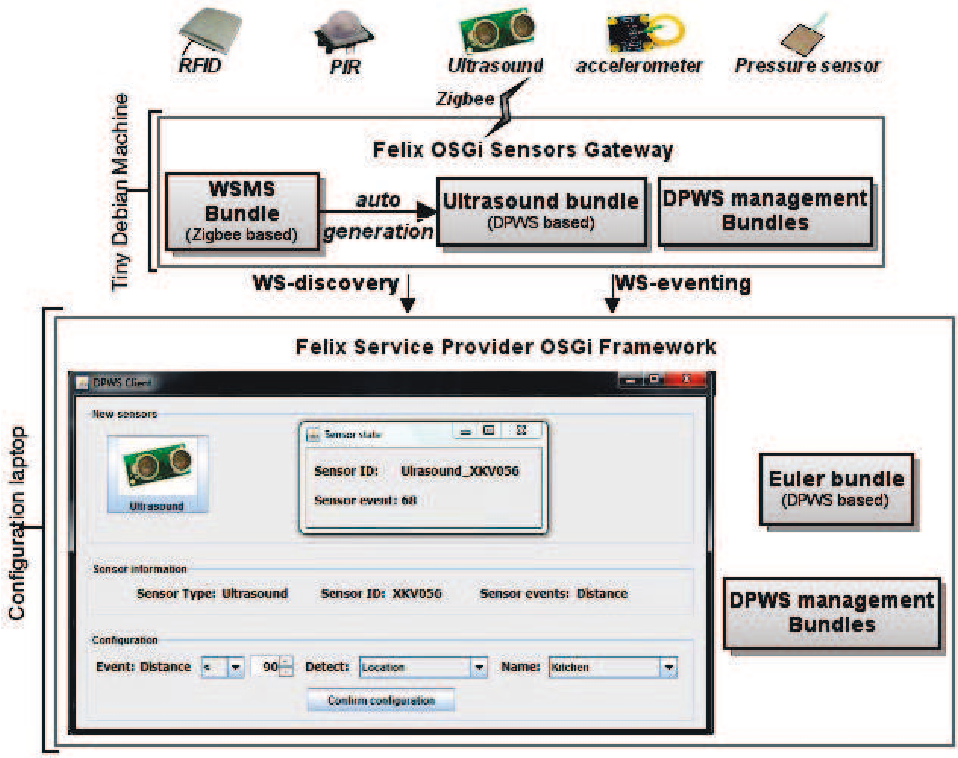


Figure 3.14: Ultrasound sensor Plug&Play scenario

sensors. I have calculated the response time needed due to the dynamic aspect of the framework. The experiment starts when the sensor is turned on in the environment and ends when it is turned off. The time for semantic enrichment is calculated, however, the time for the manipulation of the configuration tool is not integrated as it differs depending on the user. Table 3.1 illustrates our results. In both the static and dynamic configurations, an average time of 224ms is needed to start the real sensor and for the ZigBee communication required to detect the sensor presence by the framework. In addition, for the dynamic configuration, we have observed an additional average time of 413ms needed to represent a new sensor as a service in the framework. This is the time required to generate and start the bundle representing the ultrasound sensor on the gateway and to update the N3 representation of the environment with the sensor information. The average time to stop and remove a sensor representation from the gateway was about 74ms. These results represent the time difference between the static and dynamic configuration to start and stop the sensors. The time difference looks acceptable when we know that it allows adding the dynamic aspect of the framework.

Moreover, the time for context understanding, service selection and service rendering

Table 3.1: Sensor Plug&Play average time in milliseconds

Sensor Type	Sensor recognition time	Sensor removing time
Ultrasound	373	85
PIR	430	76
Accelerometer	437	61

is the same in both the static and the dynamic configuration. Table 3.2 illustrates the results.

Table 3.2: Context understanding and service provision response time in seconds

	Context	Selection	Rendering
Response time	1.2	0.8	0.7

Our approach adds the ability to integrate new assistive services with their related sensors and interaction devices at runtime. As an example, to add the new assistive service *"Wandering at night"*, we integrate its appropriate bundle in the framework then we attach a PIR sensor to the room ceiling and we put a pressure sensor under the resident mattress. The semantic Plug&Play mechanism allows integrating the service and linking its related sensors. Using the configuration tool, the PIR sensor is configured to detect the resident moving in the bedroom and the pressure sensor is configured to detect his presence on the bed. A set of first order logic rules are generated for the wandering at night service. During the night, if these rules are verified during a fixed period of time, then the resident needs an assistance to solve his wandering problem.

$$\forall \; Sensor\ s1;\ Sensor\ s2$$
$$(s1\ hasType\ PIR) \wedge (s1\ deployedIn\ bedroom) \wedge (s1\ hasCurrentState\ on)$$
$$\Rightarrow (resident\ detectedIn\ Bedroom)$$
$$(s2\ hasType\ Pressure) \wedge (s2\ deployedIn\ bedroom) \wedge (s2\ hasCurrentState\ silent)$$
$$\Rightarrow (resident\ believedToDo\ notInTheBed)$$
$$(resident\ detectedIn\ Bedroom) \wedge (resident\ believedToDo\ notInTheBed)$$
$$\Rightarrow (resident\ believedToDo\ wanderingAtNight)$$

Load test experimentation for the validation of the semantic Plug&Play approach has been realized through a progressive augmentation of the number of sensors detected and integrated into the framework at runtime. I have simulated the detection of new sensors, the generation of their corresponding bundles, and the update of the semantic representation with the newly detected sensors descriptions. The experimentation was realized on a Windows XP machine running on an Intel Core i7 CPU, with 4GB of memory. I have analyzed the augmentation of the semantic reasoning response time during the integration of new sensors. I have used two semantic rules for this experimentation. The transition from 1 sensor to 269 sensors caused an increase in the number of N3 triples from 267 to 3000. The memory limit was reach with 269 sensors. As shown in Figure 3.15, the regression function of the reasoning response time shows a linear augmentation from 370 milliseconds to 1100 milliseconds. We consider that the response time is acceptable even with 269 sensors plugged to the framework. The linear increase of the semantic reasoner response time is also a very positive result compared to other reasoners which are estimated to show an exponential shape [75].

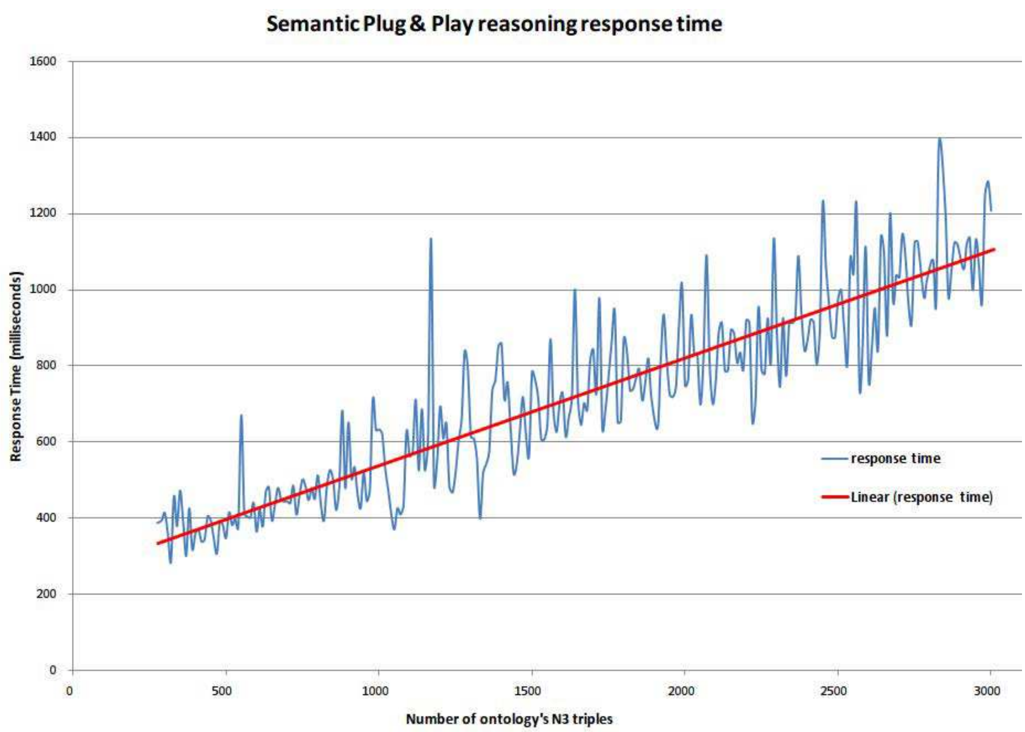


Figure 3.15: The semantic Plug&Play reasoning response time during load test

3.6 Conclusion

The realization of the UbiSMART framework offered the ability to plug new assistive services, detect and integrate new sensing technologies and interaction devices, and configure them to be functional in the assisting environment. The semantic Plug&Play approach automates the process of sensors and interaction devices integration and only a small configuration is needed to bind these entities to the assistive service to be provided.

Our real life deployment of the UbiSMART framework has identified different technical problems such as network problems and communication issues and sensors failures. Therefore, we have considered that another aspect needs to be tackled in the dynamic assistive service provider framework to cope with these problems mainly the uncertainty of information received from the environment. This aspect will be discussed in the next chapter.

4

Uncertainty Handling

4.1 Introduction

Sensory contextual information is not always accurate in the real-world. Therefore, a context-aware system should be sensible to this lack of accuracy in order to take the correct decisions. To account for this, certainty/uncertainty measurement and representation appears as a crucial need to perform a reasonable and logical reasoning. This becomes particularly important if the system using the contextual information intends to perform data fusion or higher-ordered reasoning, as is the case in our approach.

In this chapter, I present the four steps I went through in order to integrate the notion of uncertainty in our framework. Mainly the uncertainty measurement, the semantic modeling, the semantic reasoning and the decision making under uncertainty. For each section, I describe the related work then I provide my approach to cope with the requirement and its integration in the UbiSMART framework. After describing the different steps of uncertainty integration, I provide the validation of our uncertainty handling approach. I end the chapter with the final design of the UbiSMART framework taking into account the new modules integrated for uncertainty handling.

4.2 Uncertainty Measurement

Quantitative measurement and numeric representation of uncertain phenomena are hard to realize and identify. Some works have already contributed to this field for the identification of uncertainty in context information. The concept of Quality of Context (QoC) is introduced in [83, 84] as *"any information that describes the quality of information that is used as context information"* with different parameters characterizing it. Mainly the precision defined as *"granularity with which context information describes a real world situation"*, probability of correctness presented as *"the probability that an instance of context accurately represents the corresponding real world situation, as assessed by the context source, at the time it was determined"* and the up-to-dateness describing the age of the context information and defining *"how well a formerly provided context information still accurately describes the actual situation"*. This vision is applied for privacy protection using a QoC aware privacy policy framework [84]. Quality of Context is also measured based on two other parameters *"accuracy"* and *"completeness"* in [85]. However, authors do not describe any strategies to use these attributes by context-aware applications to deal with uncertainty. Bu et al. [86] have founded their work on context inconsistency resolution. The idea is to initially detect conflicting situations (such as a person located in two rooms at the same time) and then calculate the frequency of each context and eliminate the ones with smaller frequency value. A similar strategy is used in [87] with the identification of context patterns that can generate uncertainty then assess each of the contradictory context patterns and modify or eliminate the context if any matches. The last strategy requires the identification of contradictory context patterns, inconsistency on context or context-contradictory situations, which in a real scenario can be a complex task. Furthermore, contradictory situations can be numerous, making it impossible to implement in devices with limited computational resources.

The proposed solutions mentioned above are mainly based on users' habits and defined conflicting scenarios without taking into account the uncertainty generated from the hardware condition and its functional characteristics. In my approach, I am more interested in the measure of uncertainty starting from the sensing technology hardware characteristic and functional behavior perspective. I believe that such an approach will produce quantitative and numeric value on which we can base our reasoning under uncertainty to determinate the current context. This approach could be matched with the user-centered uncertainty estimation to produce more accurate measures.

Our approach is based on the identification of hardware characteristics and functional parameters that characterize the amount of confidence given to each specific sensor. For example, scrutinizing our on-site experimentation, we have noticed that most of the hardware failures occur after some period of time, when the sensors' batteries level are low, and with some level of reliability. I define these parameters as hardware characteristic parameters. It has also been perceived that these entities do not follow fixed functional behaviors over the time; such parameters have been called functional parameters. I have therefore identified the following equation to measure the amount of uncertainty derived from hardware characteristics and functional properties for each sensor:

$$M_s = \prod_{\mathbb{F}} f_i(x_i) \;\times\; \prod_{\mathbb{C}} \frac{y_j}{N \times exp(y_j/100)}$$

Where:

* $\mathbb{F}$ is a set of functional parameters that describe the way each specific type of sensors should behave.

* $\mathbb{C}$ is a set of independent characteristic parameters that define each sensor state at a specific moment.

* x_i is the value taken by the i^{th} parameter of $\mathbb{F}$
* y_j is the value taken by the j^{th} parameter of $\mathbb{C}$ (as a percentage)
* $f_i : x_i \in \mathbb{F} \longmapsto [0, 1]$ is a specific function for each functional parameter defining how much the sensor current comportment is identical to the intended behavior.
* N is a normalization factor in order to have $0 \preceq \dfrac{y_j}{N \times exp(y_j/100)} \preceq 1$

As in the case of the characteristic parameters, for which the confidence is higher when these parameters have high values and decreases sharply when they are near zero, the sensor characteristic function $\dfrac{y_j}{N \times exp(y_j/100)}$ is defined to have a shape that meets this description as shown in Figure 4.1

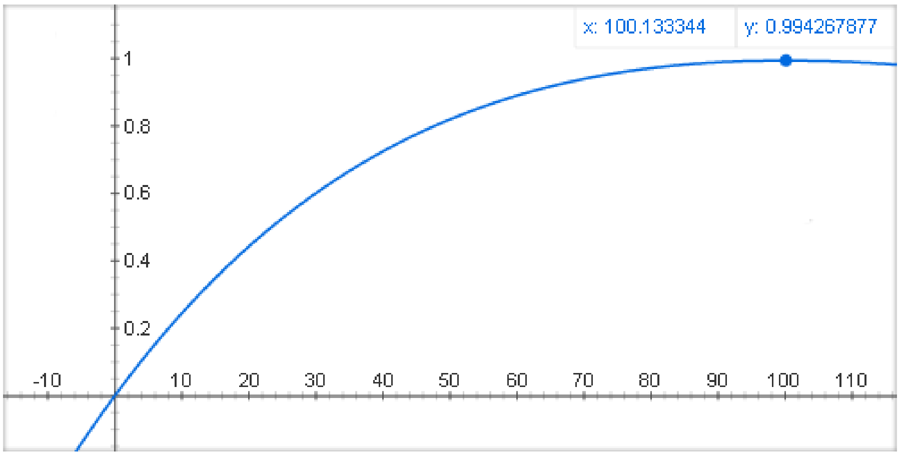


Figure 4.1: Hardware characteristic function for confidence measurement

The f_i functions are related to each type of sensor and to the uses case it will be used for. Therefore, they are defined according to the use-cases and should respect the fixed conditions above.

In order to integrate this mechanism into the UbiSMART framework, different modules have been added under *"SensApp"* to provide the measure of uncertainty about sensors needed. A central module *"Uncertainty Measure"* under *"SensApp"* receives the sensors events, then forwards them to different *"Functional Modules"* according to the type of sensor. *"Functional Modules"* calculates the functional parameters then sends them back to the central module *"UncertaintyMeasure"* that is in charge of gathering all these values and providing the output of the defined equation. The output value is combined with the sensor state and characteristics, then it is sent through the DPWS communication mechanism to the *"Semantic Module Updater"*.

4.3 Semantic Modeling under Uncertainty

The QoC is crucial when handling context information in AAL environment. Imprecise or incorrect information can engender critical situations that may affect the safety and serenity of the assisted person. Ontological knowledge is naturally processed as an absolute truth if no notion of uncertainty or QoC is introduced in the semantic model or if the reasoner is not conceived to consider these notions. In the context-aware domain, it is often the case that values coming from sensors in the environment carry with them a level of certainty in the form of numeric measures. A semantic modeling language with an alternative context representation at different levels of abstraction can cope with this by introducing classes of information and associations in accordance with these measures. This becomes particularly important if the system using the data intends to perform data fusion or higher-ordered reasoning as it is the case in context awareness [88].

Some attempts for uncertainty integration into semantic models have been realized. Hybrid's models combined with fuzzy logic [89, 90, 91], Bayesian network [92], probability [93] or Dempster-Shafer Theory [94] have been proposed. However, these models suggest a complete re-design of the used ontologies in order to fit with them.

We have separated the semantic web representation and reasoning from decision making under uncertainty. I have chosen to use the semantic web only for the integration of uncertainty representation into the semantic model then for the propagation of uncertainty to high-level context information. This keeps the same basic and classic model for context representation and allows making changes only when we need to integrate a notion of uncertainty. As soon as a decision about a conflicting and uncertain context needs to be taken, it is realized by a separate module then later updated into the ontology.

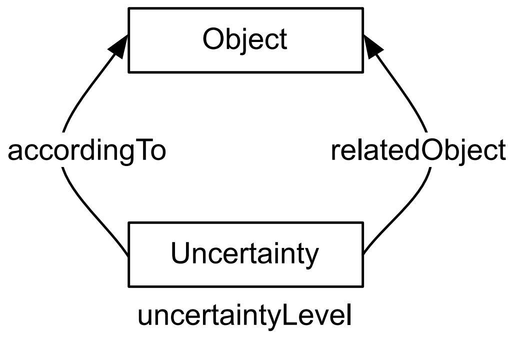
The representation of knowledge in semantic web, using RDF notion, is based on triples $\prec subject, predicate, object \succ$. It is very complicated to represent uncertainty attached to the subject or the object of the triple. As far as the subject is concerned, it may be related to different predicates, therefore, different instances representing uncertainty should be created, and each one should indicate the predicate to which it is related. The same problem happens when uncertainty is represented on the object side as it could be the target of different predicates. I considered that semantic representation of uncertainty will be the simplest and semantically logical if it is attached to the predicate. Therefore, I have provided a model for the representation of uncertainty based on some notions of

RDF mainly the *"reification"* [95] combined with the notion of *"blank nodes"* called also *"anonymous resources"* [96]. Our abstract model is defined as follow:

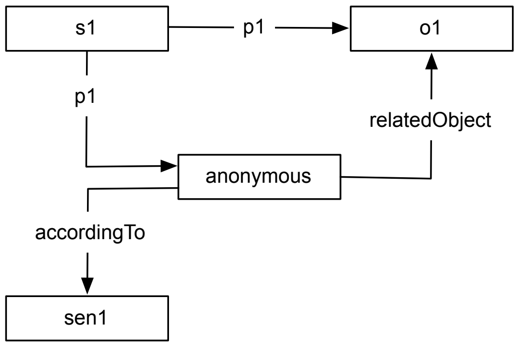
```
### MODEL ###
  unc:Uncertainty a rdfs:Class.
  unc:relatedObject a rdf:ObjectProperty;
    rdfs:comment ''Define the original range of the property.''@en;
    rdfs:domain unc:Uncertainty.
  unc:accordingTo a rdf:ObjectProperty;
    rdfs:comment ''Define the source of uncertainty.''@en;
    rdfs:domain unc:Uncertainty.
  unc:uncertaintyLevel a owl:DatatypeProperty;
    rdfs:comment ''define the the property degree of uncertainty.''@en;
    rdfs:domain unc:Uncertainty;
    rdfs:range xsd:double.
```

I integrate a new class *"Uncertainty"* into the model with two object properties and one data property. An anonymous resource will be instantiated from *"Uncertainty"* each time a notion of uncertainty needs to be expressed. This anonymous resource is related to the *"subject"* through the *"predicate"* and has the data property *"uncertaintyLevel"* representing the level of uncertainty that the *"predicate"* is linking the *"subject"* to the *"object"*. The anonymous resources is also linked to the *"object"* through the property *"relatedObject"*. The property *"accordingTo"* is used to link the anonymous resource to the source of uncertainty, meaning the sensor or the set of sensors from which the uncertainty about $\prec subject, predicate, object \succ$ derive. Figure 4.2 shows the uncertainty model and its possible instantiation.

The N3 syntax representing the transition from a classic triple to a triple including uncertainty is given below:



(a) Model



(b) Instantiation

Figure 4.2: Uncertainty model and instantiation

```
## Classical Presentation
hom:s1 a aal:C1.
hom:o1 a aal:C2.
hom:p1 a rdf:ObjectProperty;
   rdfs:domain aal:C1;
   rdfs:rang aal:C2.
hom:s1 aal:p1 hom:o1.
```

```
## Presentation including Uncertainty
hom:s1 a aal:C1.
hom:o1 a aal:C2.
hom:sen1 a aal:Sensor.
hom:p1 a rdf:ObjectProperty;
   rdfs:domain aal:C1;
   rdfs:rang aal:C2.
hom:s1 hom:p1 [ a unc:Uncertainty;
   unc:uncertaintyLevel 86.4;
   unc:relatedObject hom:o1;
   unc:accordingTo hom:sen1].
```

This approach for semantic representation of uncertainty has many advantages. It allows to keep the use of the classical representation and integrates the notion of uncertainty only when it is needed. Therefore, our resulting ontology will have some parts with classical representation, while other parts include uncertainty representation.

4.4 Reasoning under Uncertainty

The levels of certainty received should be propagated through the reasoning in order to achieve the representation of high-level-context uncertainty. In other words, when the firing of a semantic reasoning rule results in the inference of a high-level context, its certainty level needs to be calculated from the certainty levels of the values used in the antecedent of the rule [97]. This approach has been adopted in our framework, where the uncertainty is propagated from sensors level to high-level context information using semantic rules. Next, I provide the transition of a semantic rule from a classical representation to a new one that transfers uncertainty from sensors level to context level. This rule is used for localization tracking of the end-user.

Classical representation:

$\forall$ *Sensor se; SensorState st; Room r; House h; User u*

(se hasCurrentState st) $\wedge$ *(se hasLastUpdate true)* $\wedge$ *(se deployedIn r)*

$\wedge$ *(r partOf h)* $\wedge$ *(u liveIn h)* $\Rightarrow$ *(u detectedIn r)*

Uncertainty inclusion:

$\forall$ *Sensor se; SensorState st; Room r; House h; User u*

(se hasCurrentState [a Uncertainty; uncertaintyLevel n; relatedObject st])

$\wedge$ *(se hasLastUpdate true)* $\wedge$ *(se deployedIn r)* $\wedge$ *(r partOf h)* $\wedge$ *(u liveIn h)*

$\Rightarrow$ *(u detectedIn [a Uncertainty; uncertaintyLevel n; relatedObject r; accordingTo se])*

We believe that the uncertainty aspect will not be tackled only by the engine itself, but it is rather the way the engine is used and coupled with other techniques that can ever address this aspect. Especially when aiming at data fusion or higher-ordered reasoning in AAL systems.

Diverse formalisms exist to deal with uncertainty including probability value assignments, and degrees of set membership for vagueness e.g., probabilistic approach [98], Bayesian reasoning [92], Dempster-Shafer techniques [99], and fuzzy logic [100]. These mechanisms can be coupled with semantic web reasoning.

4.4.1 Probabilistic Approach

Probability theory is the first uncertainty management technique to be introduced. This theory seeks to judge the probability measure for an event A_i given a proposed hypothesis H_i such that:

$$0 \leq Pr(A_i|H_i) \leq 1$$

Decision making in probabilistic approach could be realized using different rules. For example, the likelihood comparison rule suggests accepting the hypothesis H_i if the probability relationship satisfies the equation:

$$P(A_k|H_i).P(H_i) > P(A_k|\neg H_i).P(\neg H_i)$$

One principal limitation of the traditional probability theory's characterization of uncertainty is its incapability of capturing epistemic uncertainty. The application of traditional probabilistic methods to epistemic or subjective uncertainty is often known as Bayesian probability [99]. In addition, a probabilistic analysis requires an analyst to have information on the probability of all events. When this is not available, the uniform distribution function is often used to affect an equal value to all events for which a probability distribution is not known in a given sample space. This is not totally true. For example, if we have three sensors in the environment, we assign a probability of failure with 0.4 to one of them and we are ignoring the probability of the two others. This does not mean that they have a probability of failure of 0.3 each. Another assumption in classical probability is that the knowledge of the likelihood of the occurrence of an event can be translated into the knowledge of the likelihood of that event not occurring. Once again, if we believe that the system may fail due to the first sensor with a probability of 0.4, this does not necessarily mean that we believe that the system will not fail due to that sensor with a probability of 0.6.

Though the assumptions of additivity and the principle of insufficient reason may be appropriate when modeling the random events associated with aleatory uncertainty, these constraints are questionable when applied to an issue of knowledge or belief, especially when information on which to evaluate a probability is limited, ambiguous or conflicting. It could be reasonable to consider probability measurement as an interval or a set when a precise probability characterizing the uncertainty is not available. This has three major advantages:

- We do not have to give a precise measure of uncertainty if it is not realistic or feasible to do so.
- Uncertainty could be affected to multiple events together without having to give assumption about events under ignorance.
- Measure of uncertainty does not have to add up to 1. If it is less than 1, this means that there is incompatibility between multiple sensors providing conflicting information. If it is greater than 1, this implies a cooperative effect between multiple sensors providing the same information.

Three different approaches can be applied to address the problem of interval-based uncertainty representation: imprecise theory, possibility theory and the Dempser-Shafer Theory (DST) of evidence.

4.4.2 Bayesian Inference

The Bayesian inference came to solve some limits of the classical probabilistic approach. It consists mainly in updating the probability of a hypothesis given previous probability estimation.

The Bayesian inference calculates the probability $P(H_i|A)$ that the hypothesis evidence H_i has occurred given an observed event A using the following equation [101]:

$$P(H_i|A) = \frac{P(A|H_i).P(H_i)}{\sum_j P(A|H_j).P(H_j)}$$

Where P(H_i) is the probability that the hypothesis evidence or event has occurred; and P(A | H_i) is the probability that the evidence A is observed given the hypothesis H_i has occurred. The basic Bayesian inference was extended to take into consideration the notion of time through the Dynamic Bayesian network (DBN). A DBN is mainly a set of Bayesian networks copies for different moments with conditional dependencies between past and future copies [81]. It can be represented by the equation:

$$P(X_{1:T}) = \prod_{t=1}^{T} \prod_{i=1}^{n} P(X_i^t|\pi(X_i^t))$$

Where X is a variable for moments ranging from 1 to T.

The use of the Bayesian inference method could be beneficial as it allows to incrementally estimate the probability of a hypothesis being true, given new observation. It also allows incorporating a priori knowledge about a hypothesis in the inference process. However, using this approach, it is difficult to define a priori probabilities. In addition, it requires mutual exclusivity for competing hypotheses and become complex when there are mutual potential hypotheses and multiple conditionally-dependent events. Finally, this approach is unable to account a general uncertainty based on multiple hypotheses [99].

4.4.3 Possibility Theory: Fuzzy Logic

The theory of possibilities has been first introduced by Zadeh [102] using a notion of fuzzy set [103]. In this theory, the uncertainty about an event consists of a subjective measure of surprise or possibility an observer will express if the event occurs. In that sense, if an event is entirely possible, then there is no surprise attached to its occurrence. However, if it is impossible, then if it occurs it will be assigned the maximum degree of surprise [104]. The possibility theory is based on the notion of fuzzy set with non absolute membership. A membership function μ_F specifies the degree to which each object in the universe is a member of the fuzzy set F. Possibility theory contradicts the classical probabilistic approach in two principal points:

$$A \cup \overline{A} \neq X \text{ meaning } \mu_{A \cup \overline{A}}(x) \neq 1$$

$$A \cap \overline{A} \neq \emptyset \text{ meaning } \mu_{A \cap \overline{A}}(x) \neq 0$$

Despite the advantages provided by the possibility theory, there are still some problems related to its application for context understanding. Mainly the need for fuzzy sets identification which requires the implication for experts or learning process, and its aspect of dealing with vague statements which is beneficial for human based reasoning on the contrary of context aware reasoning based on sensors information and machine reasoning.

4.4.4 Evidence Theory: Dempster-Shafer Theory

Also called the Dempster-Shafer theory to refer to the work realized by Arthur Dempster [105, 106] and Glenn Shafer [107] in order to overcome limitations of the probability theory and specially the completeness axiom. In DST, an evidence can be associated with multiple possible events (set of events) on the contrary of classical probability approach. It results in meaningful evidence at high level of abstraction without having to resort to assumptions about the events. In that sense, the DST model allows to cope with varying levels of precision regarding the information without recourse to assumptions to represent the information. The representation of uncertainty for an imprecise input can be characterized by a set or an interval [108].

DST is based on a set of non dividable, mutually exclusive and exhaustive hypotheses called *the frame of discernment* θ. From the frame of discernment is generated the power set Θ which is a set of subsets of θ and includes all the possible combinations of its elements. Each element from the power set Θ can be assigned a value called *basic probability assignment* or *mass function*.

$$m : \Theta \longrightarrow [0, 1]$$

In order to prohibit committing any confidence mass to an empty hypothesis, the mass function should satisfy the condition:

$$m(\emptyset) = 0$$

The mass function measure for a specific hypothesis H does not include the measure of mass function of any subset of that hypothesis. If $H = \{w_1, w_2\}$, $m(H) = 0.7$ represents only the amount of confidence in that either w_1 or w_2 is true and does not give any information about the measure of confidence for w_1 or w_2 alone. In addition, the DST allows assigning mass functions to hypotheses from Θ that we have idea about their confidence and to ignore assignments to those we know nothing about. A measure of ignorance can therefore be assigned to the maximum uncertain hypothesis θ through the mass function $m(\theta)$ representing the amount of confidence we are unable to assign, through lack of knowledge, to any subset of Θ. If only the mass function $m(w_1, w_2) = 0.7$ is known, the remaining mass 0.3 is assigned to θ using the following condition that should be satisfied in DST [109]:

$$\sum_{H \in \Theta} m(H) = 1$$

Based on this, committing a mass to a hypothesis does not imply committing the remaining mass to its negation as it is the case in classical probability.

$$m(H) = q \nRightarrow m(\overline{H}) = 1 - q$$

The belief in a hypothesis H from θ is the sum of the mass of all its subsets. It is the sum of all the hypothesis mass that expresses the total confidence in the truth of the hypothesis H and its entire constituent:

$$Belief(H) = \sum_{h \subset H} m(h)$$

Similarly, the plausibility function is the degree to which the hypothesis is possible and is defined. It is the sum of all the mass functions of the sets that intersect the set of the hypothesis H:

$$Plausibility(H) = \sum_{H' | H' \cap H \neq \emptyset} m(H')$$

DST has the ability to combine uncertainty emerging from different sources. The theory of Dempster-Shafer is based on the principal of observations. Therefore, if many observations are available, it has the ability to combine all of them in order to get the most realistic and consensual uncertainty value. The Dempster-Shafer Evidence combination rule, called the joint mass function, provides a means to combine these observations. For two pieces of evidence about a Hypotheses H in the form of mass functions m_1 and m_2, the combined mass function m_c is calculated using the Dempster-Shafer combination rule:

$$m_c(H) = \frac{\sum\limits_{\forall X,Y:X\cap Y=H} m_1(X).m_2(Y)}{1 - \sum\limits_{\forall X,Y:X\cap Y=\emptyset} m_1(X).m_2(Y)} \text{ when } H \neq \emptyset$$

$$m_c(\emptyset) = 0$$

The numerator of the Dempster-Shafer combination rule sums the mass function products of the hypothesis whose set intersection is exactly the hypotheses H. The denominator 1-K where $K = \sum\limits_{\forall X,Y:X\cap Y=\emptyset} m_1(X).m_2(Y)$ is a normalization factor allowing to assign a null mass function to all empty set intersection hypotheses and redistribution the resulting excess mass function among the non-empty hypothesis by the factor 1-K where K is the total of combined mass functions corresponding to the empty hypotheses. Meaning that we are eliminating all the impossible combinations where the confidence to an empty set is different from zero.

The Dempster-Shafer combination rule is both associative and commutative [108]. In fact, $m_1(X)$ can be the result of a previous combination rule, so that the process of combining evidence from multiple sources can be chained and the order in which the sources are combined does not affect the final result [99].

4.4.5 Decision-Supporting Mechanism

I have chosen to use the Dempster-Shafer theory which has relatively high degree of theoretical development among the non-traditional theories for characterizing uncertainty and has a versatility to represent and combine different types of evidence obtained from multiple sources. In fact, DST is chosen as the first core module to implement the sensor fusion algorithm. It is shown to provide a sensor fusion performance advantage over previous approaches, e.g., Bayesian Inference approach, as it can better imitate human uncertainty-handling and reasoning process [99].

The DST implementation is used as a support for our reasoning engine. After the reasoning process with uncertainty, the *"Thought Interpreter"* module invokes a new integrated module called *"DST Decision Making"* which manages the different mass functions for possible hypothesis. Each mass function is related to a set of sensors deduced from the *"accordingTo"* property and to a specific context of the end-user which is in fact the basic property of the fired rule. The mass function value is the uncertainty measurement deduced from the *"uncertaintyLevel"* property. Possible hypotheses are all the instances of the object class of the fired triple, deduced from the *"relatedObject"* property. For conflicting context situations, the Dempster-Shafer combination rule is used to combine information raising from different sensing sources and the decision on the current context of the end-user is taken based on the output of the combination rule.

4.5 Validation of the Uncertainty Handling Approach

For the validation of our approach of uncertainty handling, we take the example of location tracking and activity recognition. Motion sensors are deployed in different rooms of the patient's house in order to detect his location. Due to faulty hardware or communication problems, we sometimes receive conflicting information about the user location. I have set three characteristic parameters: *"battery's level"*, *"reliability"* and *"sensor lifetime"*. In addition, as a functional parameter I have defined a characteristic function that represents the normal behavior of the sensor. Battery level differs from one sensor to another even though sensors are deployed at the same time. This is mainly due to the use of the sensors and the transmission of events, which consumes cruelly the batteries. Figure 4.3 shows the evolution of the battery levels of different deployed sensors within a month.

The reliability parameter was calculated based on experiments on the deployed sensors where we specify a succession of events to be sent with latency between each two events and we calculate the number of events lost. Table 4.1 represents the results for the different sensors.

The sensor lifetime parameter represents the percentage of the duration since the sensor has been deployed with respect to the estimated lifetime. For the functional parameter, which is use-case related, we have used the following reasoning. Once deployed in a room, motion sensors are sending continuously *"on"* events with a latency of 10 seconds while the patient is moving. If he stops moving for 60 seconds, the sensor sends an *"off"*

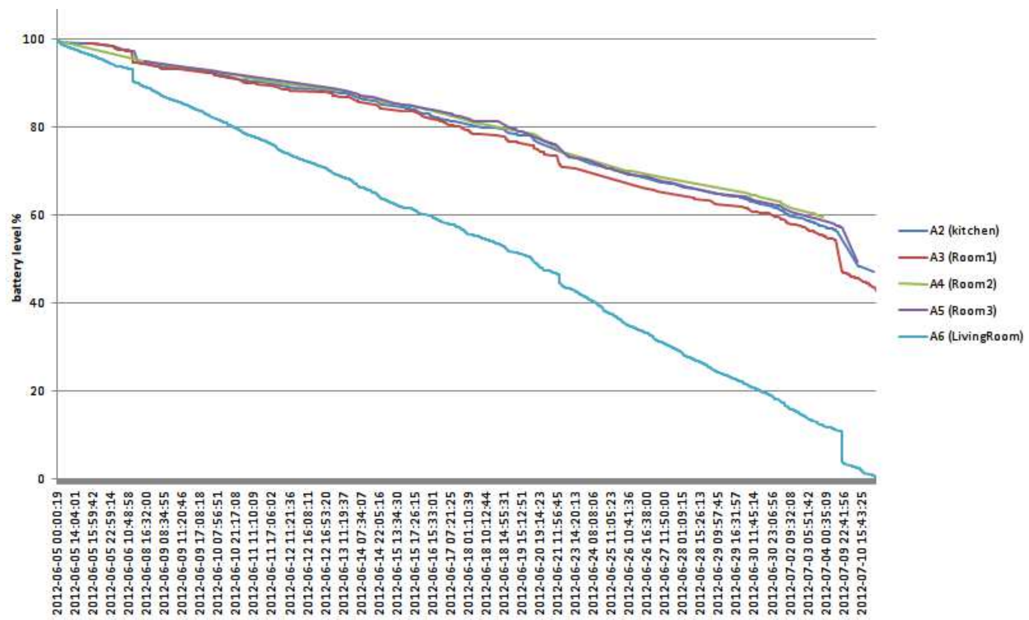


Figure 4.3: Deployed sensors batteries' level over the time

Table 4.1: Deployed sensors' reliabilities

Sensor	Total events sent	Total events received	Reliability
A2 (kitchen)	1106	758	68.53%
A3 (Room1)	1834	1526	83.2%
A4 (Room2)	276	237	85.87%
A5 (Room3)	528	501	94.88%
A6 (LivingRoom)	10658	10213	95.82%

event and switches to a standby mode until the patient moves again. For each sensor, I have calculated the maximum number of *"on"* events (maxevent) that can be received successively and considered that the confidence to sensor start decreases beyond these values. In order to realize this reasoning, I have defined the following equation which keeps the functional parameter around 100% while the total of successive *"on"* events is less than the maximum, then abruptly decreases toward 0% when we outstrip the limit.

$$F(x) = 100 - \left(\frac{x \times N}{m \times maxevent}\right)^i \text{ if } x \leq (m \times maxevent)$$

Where:

* "i" is an exponent factor which defines the shape of the function. In our case, I have

chosen i=6, which represent well our intended behaviour of the function described above.

* "x" represents the number of successive events sent by the sensor up to the current event.

* "m" is a multiplication factor of maxevent. The result defines the number of successive events after which the functional parameter confidence will be equal to 0. In our case I have fixed the value of m to 3/2.

* "N" is a normalisation factor in order to have F(x)=0 when x = $m \times maxevent$. In this example, N = 2.16.

Figure 4.4 represents the shape for this equation for a sensor having maxevent set at 164 while Table 4.2 represents the maxevent values calculated for each sensor.

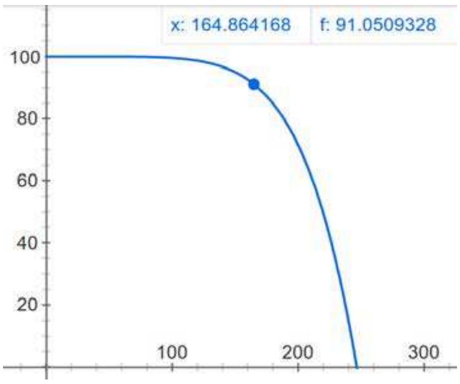


Figure 4.4: Functional equation shape for maxevent = 164

Table 4.2: Deployed sensors' maxevent values

	A2 (kitchen)	**A3 (Room1)**	**A4 (Room2)**	**A5 (Room3)**	**A6 (LivingRoom)**
MaxEvent	52	38	21	61	164

All the parameters have been used to measure the mass function of the different sensors, based on the equation for uncertainty measurement defined in section 4.2. The results for the different sensors are presented in Figure 4.5. The different parameters impact on the mass functions is visible. We can see for short moments sharp decreases of these values related to the functional parameter results.

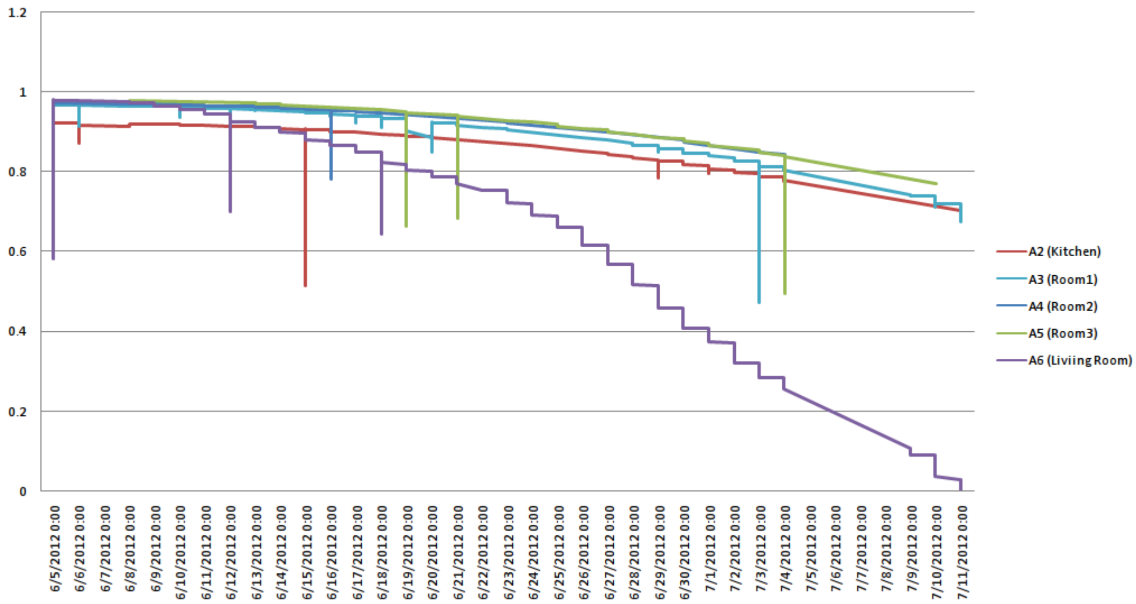


Figure 4.5: Sensors' mass functions results over the time

In order to infer the user location context using the semantic rules, and taking into account the amount of uncertainty associated to each sensor, we have defined the following first order logic rule:

$$\forall \; Sensor\ se;\ SensorState\ st;\ Room\ r;\ House\ h;\ User\ u$$
$$(se\ hasCurrentState\ [\ a\ Uncertainty;\ uncertaintyLevel\ n;\ relatedObject\ st])$$
$$\wedge\ (se\ hasLastUpdate\ true) \wedge (se\ deployedIn\ r) \wedge (r\ partOf\ h) \wedge (u\ liveIn\ h)$$
$$\Rightarrow (u\ detectedIn\ [\ a\ Uncertainty;\ uncertaintyLevel\ n;\ relatedObject\ r;\ accordingTo\ se])$$

Lets take the example of a conflicting contextual situation where the two sensors *"A4"* and *"A6"* are firing at the same time. When the semantic rule is triggered, two mass functions m_detectedIn_a4 and m_detectedIn_a6 are created. The two mass functions produce the Table 4.3.

This information gives an idea about the possible activity. We define a set of activities that we want to infer: sleeping, showering, wanderingBedroom, wanderingLivingroom and watchTV. We therefore get two mass functions corresponding to the property *"believedToDo"* as shown in Table 4.4.

In order to have common information emerging from the different sensors, we apply the Dempster-Shafer combination rule which produces the results in Table 4.5.

Here we can confirm that the patient is performing activities in the bedroom, but we cannot confirm which activity he is performing. After some seconds, we receive events

Table 4.3: Mass functions of two sensors for location tracking

Context/detectedIn	A4 mass function	A6 mass function
livingRoom	0.0	0.743
kitchen	0.0	0.0
room1	0.0	0.0
room2	0.923	0.0
room3	0.0	0.0
ignorance	0.077	0.257

Table 4.4: Mass functions of two sensors for activity recognition

Context/believedToDo	A4 mass function	A6 mass function
sleeping	0.0	0.0
showering	0.0	0.0
wanderingBedroom	0.0	0.0
wanderingLivingroom	0.0	0.0
watchTV	0.0	0.0
sleeping,wanderingBedroom	0.923	0.0
watchTV,wanderingLivingroom	0	0.743
ignorance	0.077	0.257

from a pressure sensor deployed on the sofa in the living room which has a confidence of 0.741. The Table 4.6 presents the mass functions of *"believedToDo"* property taking into account the new event received.

A new Dempster-Shafer combination is realized between the three sensors that produce the results in Table 4.7:

Therefore, the patient is considered to be detected in the living room, and watching the TV. The semantic model is updated so that this information will be taken into consideration in further reasoning. The pressure sensor has provided additional information which helped to refine our reasoning and support the motion sensor in the living-room

Table 4.5: DST combination rule results for two sensors

Context/believedToDo	A4,A6 mass function
wanderingBedroom,sleeping	0.754
wanderingLivingroom,watchTV	0.128
ignorance	0.063
Context/believedToDo	**A4,A6 plausibility**
wanderingBedroom	0.818
sleeping	0.818
wanderingLivingroom	0.245
watchTV	0.245
showering	0.062

Table 4.6: Mass functions of three sensors for activity recognition

Context/believedToDo	mass functions		
	A4	**A6**	**P1**
sleeping	0.0	0.0	0.0
showering	0.0	0.0	0.0
wanderingBedroom	0.0	0.0	0.0
wanderingLivingroom	0.0	0.0	0.0
watchTV	0.0	0.0	0.741
sleeping, wanderingBedroom	0.923	0.0	0.0
watchTV, wanderingLivingroom	0.0	0.743	0.0
ignorance	0.077	0.257	0.259

which had less confidence. This case shows the importance of uncertainty fusion from different sources in order to get more precise results.

Table 4.7: DST combination rule results for three sensors

Context/believedToDo	**A4,A6,P1 mass function**
wanderingBedroom,sleeping	0.443
wanderingLivingroom,watchTV	0.117
watchTV	0.412
ignorance	0.037
Context/believedToDo	**A4,A6,P1 plausibility**
wanderingBedroom	0.48
sleeping	0.48
wanderingLivingroom	0.144
watchTV	0.556
showering	0.037

4.6 Design & Development

For the design of the UbiSMART framework including the management of uncertainty, we present in this section the components diagram of the UbiSMART framework with three sequence diagrams that represent the most important steps in the UbiSMART functioning. Namely, the "discovery and configuration of sensors", the "context update and mass functions calculation" and finally the "decision making, service selection and provision". A class diagram at the end shows the principal interfering modules in uncertainty handling and the different relations and dependencies between their classes.

4.6.1 Components Diagram

The components diagram in Figure 4.6 represents the different modules of the UbiSMART framework that have been implemented and the interaction between them. In OSGi implementation, each module exports one or more services (interfaces) which are imported and consumed by other modules.

"Sensor Management System" exposes the *"SensorController"* interface which is used by the *"Sensor Gateway"* to send sensors' events.

"Sensor_i" is an abstraction of all the sensors' modules. Each of them has an ex-

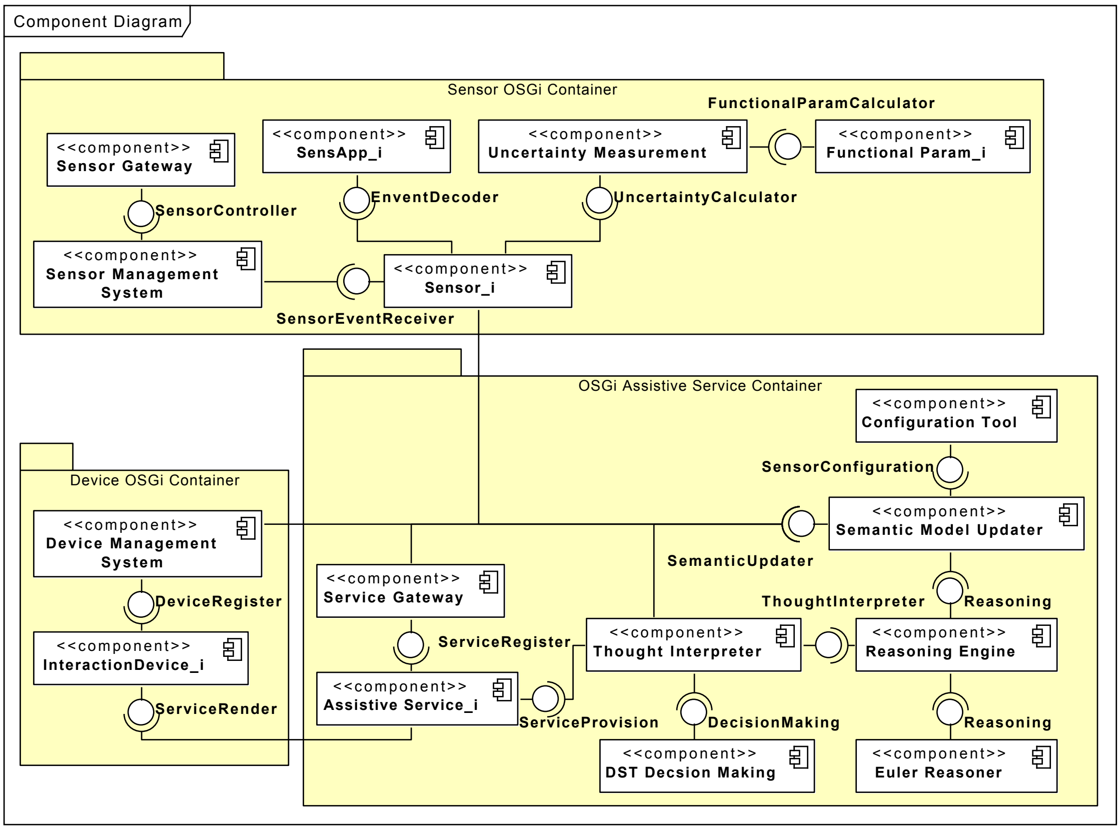


Figure 4.6: Component diagram of the UbiSMART framework

posed service *"SensorEventReceiver"* used by the *"Sensor Management System"* to send new sensors' events. *"Sensor_i"* module consumes *"EventDecoder"* service from *"SensApp_i"* to transform raw sensors' data to macro context, *"UncertaintyCalculator"* from the *"Uncertainty Measurement"* module to calculate the mass function of each sensor and the *"SemanticUpdater"* interface from the *"Semantic Model Updater"* used to update the model with the description of the sensor and its bindings.

"SensApp_i" is an abstraction of the different Sensor applications used in the framework. Each one is exposing an *"EventDecoder"* interface.

"Uncertainty Measurement" is responsible of the calculation of the mass function of each sensor through the *"UncertaintyCalculator"* interface. It invokes different functional parameter modules to integrate their results in the calculation.

"FunctionalParam_i" is the abstraction of the different functional parameters modules. Each one exposes an interface *"FunctionalParamCalculator"*.

"Semantic Model Updater" presents one interface called *"SemanticUpdater"* to receive events from the different sensors. It invokes services from the *"Configuration Tool"* to add semantic bindings to the discovered sensors and from the *"Reasoning Engine"* to start reasoning each time the context changes.

"Configuration Tool" is responsible of the semantic configuration of the new detected sensors through the *"SensorConfiguration"* interface.

"Reasoning Engine" exposes the service *"Reasoning"* used by the *"Semantic Model Updater"* and invokes the *"ThoughtInterpretor"* service to decode the result of the reasoning.

"Euler Reasoner" packs the Euler engine as a bundle and exposes the service *"Reasoning"* in order to use the engine by the *"Reasoning Engine"* module.

"Thought Interpreter" exposes the *"ThoughInterpreter"* interface to receive the output of the reasoning engine. It decodes the reasoning engine output and uses the *"DecisionMaking"* service of the *"DST Decision Making"* module to handle conflicts. The *"ThoughtInterpreter"* module updates the semantic model through the *"SemanticUpdater"* service and starts the selected service through the *"serviceProvision"* interface.

"DST Decision Making" is used to combine evidences from different resources and take decisions in case of conflicting contexts.

"AssisitiveService_i" is an abstraction of the different assistive services that can be provided. It exposes the *"ServiceProvision"* service, invokes the *"Service Gateway"* to register once installed, and invokes the *"InteractionDevice_i"* to be rendered.

"Service Gateway" is responsible of registering the different assistive services. It uses the *"semanticUpdater"* interface to register new assistive services to the semantic model.

"interactionDevice_i" is an abstraction of all the detected interaction devices in the environment. Each interaction device registers itself to the framework through the *"Device Manager System"* module using the *"DeviceRegister"* service. It exposes the *"ServiceRender"* service to render the assistive services.

"Device Management System" uses the *"SemanticUpdater"* service to update the semantic model with the newly detected devices description.

4.6.2 Sequence Diagram for Sensors Discovery and Configuration

This use case starts when a new sensor is turned on in the environment and is sending discovery events. The UbiSMART framework discovers this sensor through the *"X Gateway"* which forwards the received event to the *"Sensor Management System"*. This module decodes the received events and provides the sensor's identifications. It accordingly generates and starts a new bundle representing the new discovered sensor into the framework. The new bundle sends a DPWS discovery event to the assistive services' container. Once this event is received by the *"Semantic Model Updater"* module, sensor's information are displayed on the *"Configuration Tool"* and the sensor is configured to detect specific context. The *"Semantic Model Updater"* updates the semantic model with the sensor information and its intended use. The configuration of the new sensor is based on uncertainty, which means that the generated rules from this configuration use the notion of uncertainty in rules syntax presented earlier. This whole process is illustrated in Figure 4.7.

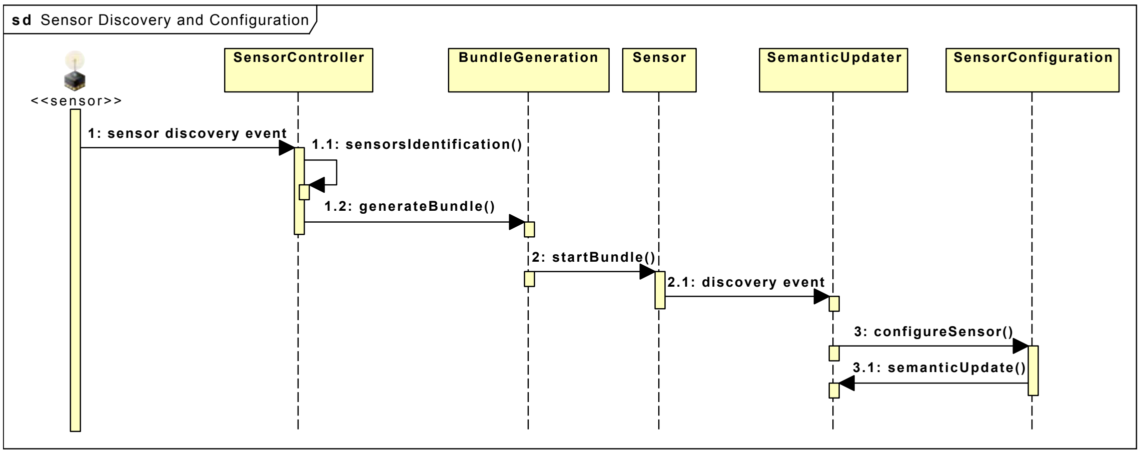


Figure 4.7: Sequence diagram for sensors discovery and configuration

4.6.3 Sequence Diagram for Context Update and Mass Function Calculation

Once the new sensor is configured for the detection of a specific context, each new event received is decoded, and the semantic model is updated. The *"Sensor Management System"* detects new events then forwards them to the appropriate sensor's bundle. *"Sensor"* uses *"SensApps"* to decode the event and sends the information to the *"Uncertainty Measurement"* module. This module invokes the *"functional Params"* module to estimate the functional state of the sensor, then measures the uncertainty based on predefined characteristic parameters. The semantic model is updated with the sensor's state and the level of uncertainty associated. The sequence diagram of the Context Update and Mass Function Calculation is given in Figure 4.8.

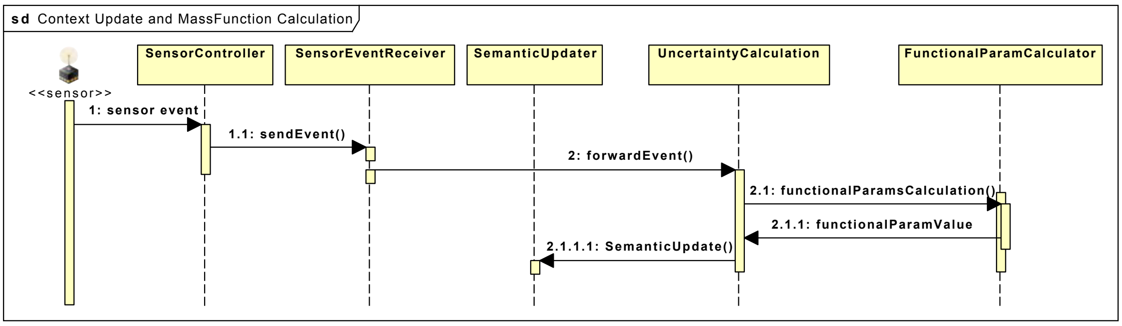


Figure 4.8: Sequence diagram for context update and mass functions calculation

4.6.4 Sequence Diagram for Decision Making and Service Selection and Provision

After each context update, the UbiSMART framework performs some processing for decision making on conflicting situation using DST, then the semantic reasoner decides on the assistive service and interaction device to use. In fact, the *"Semantic Model Updater"* launches the reasoning after each context update. The *"Reasoning Engine"* performs the reasoning and the output is provided to the *"Thought Interpreter"* module. This module asks the '*"DST Decision Making"* module to take decisions about conflicting contextual situations. Once decisions are taken, *"Thought Interpreter"* updates the semantic model, and starts the selected assistive service by the *"Reasoning Engine"*. When the *"Assistive Service"* is operated, it receives information about the device to use. Therefore, the *"Assistive Service"* module invokes the corresponding device bundle which is in charge of communicating with the real device and provides the service. The Figure 4.9 shows the sequence diagram for Decision Making and Service Selection and Provision.

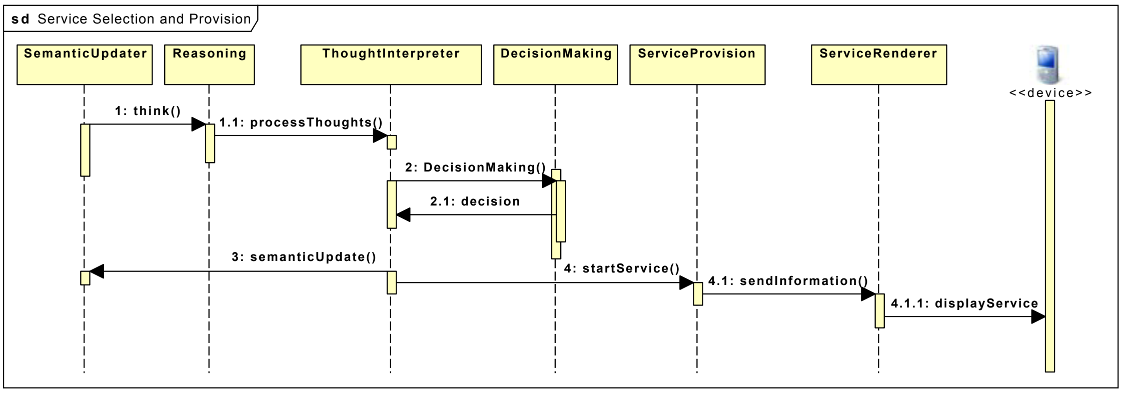


Figure 4.9: Sequence diagram for decision making and service selection and provision

4.6.5 Class Diagram

The UML class diagram presented in Figure 4.10 shows the principal interfering modules in uncertainty handling and the different relations and dependencies between their classes.

Some modules such as *“Sensor Gateway”*, *“SensApps”*, *“Functional Params”*, *“Reasoning Engine”* and *“Thought Interpretor”* are based on one class that implements the exposed interface to provide the intended service. However, other modules are composed of different classes. The *“Sensor Mangement System”* implements the *“SensorController”* interface to receive sensors’ events. *“SensorAbstraction”* class is used to get the identification of a new sensor, while *“BundleGeneration”* class is used to generate a new bundle representing a new detected sensor. A sensor module is composed of the implementation of the exposed interface *“SensorEventReceiver”*, the *“PseudoSensor”* class used to get the description of the sensor from the generated bundle, and a *“SensorRegistration”* class used to publish the new sensor description to the *“Semantic Model Updater”* on the OSGi assistive services’ container. The *“Semantic Model Updater”* uses the *“eventHandler”* method of the *“Decoder”* class to decode received sensor’s description and update the semantic model. It also implements the *“SemanticUpdater”* interface to receive sensors’ events and invoke different modules for sensor configuration and reasoning. The *“Configuration Tool”* module uses the *“Sensor”* class in order to configure different sensors by the *“SensorConfiguration”* class. Once the sensor is configured and its events are received, the *“Uncertainty Measurement”* Module uses the *“UncertaintyCalculator”* to calculate the confidence to the sensor. *“UncertaintyCalculator”* invokes the *“FiabilityCalculator”* and the *“AutonomyCalculation”* classes in the same module, and the *“FunctionalParamCalculator”* from the *“Functional Param”* module to produce the result. In the case of conflicting situations, the *“DST Decision Making”* module is used. The *“DecisionMaking”* class uses the *“DST-Calculation”* class to create the different massfunctions for all the possible hypotheses and make the fusion of the different sensor confidence to make decision. More details about the implementation of our uncertainty approach are explained in Appendix E.

4.7 Conclusion

The integration of the uncertainty handling approach provides the UbiSMART framework with the ability to manage some conflicting situations arising mainly from the sensory technologies faulty hardware and the communication problems. The integration is real-

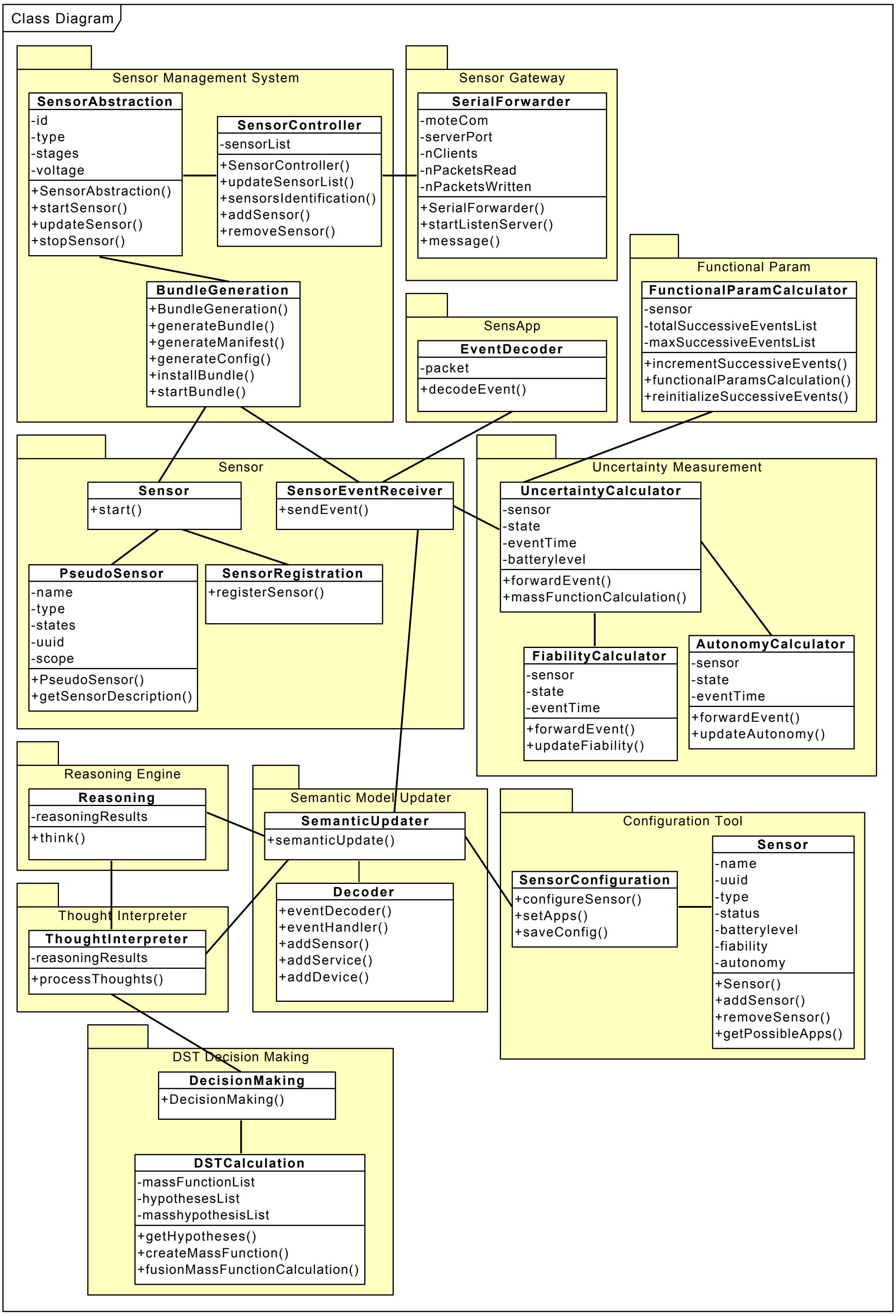


Figure 4.10: Class diagram of the UbiSMART framework

ized on the main principal layers of the framework: first, on the service model layer via the integration of new modules managing uncertainty measurement and decision making. Second, on the service interoperability layer with the reception of sensors' characteristics information and the transfer of related uncertainty values. Third, the integration was realized on the semantic reasoning layer through the semantic representation and reasoning under uncertainty.

5

Deployment and Validation

5.1 Introduction

This thesis has been realized alongside with a project for real deployment in a nursing home, which lasted for two years. The UbiSMART framework has been developed during this period with reliance to the investigations and discussions we got on-site. The framework has evolved along the project from a static configuration during the initial phase to the dynamic configuration using the semantic Plug&Play mechanism. The deployment also allowed to initiate the work achieved for uncertainty handling. A system for ambient assistive living including the UbiSMART framework has been deployed in a nursing home for elderly with mild-dementia assistance. We have focused on stages 3 to 5 of the disease, at which the deficits are typically still mild and, therefore, amenable to assistive intervention. At these stages, the patient experiences difficulties in planning, organizing and sequencing that prevent him/her from performing tasks in an ordered and sequenced manner. Distractions or a short-term memory problem may lead the patient to skip steps unwittingly or to perform actions that are unrelated to his or her original goal [110]. Monitoring and intervention when necessary can help the patient remain independent in his/her home for as long as possible [111].

Assistive services included in the UbiSMART framework are defined with the help of the professional caregivers from our collaborating nursing home. For example, the caregivers have proposed some abnormal behaviors that their patients experience such as wandering, showering without soap, showering for too long, using wrong bed, etc. Each of these behaviors is assisted by a specific service that can be integrated into the framework. The dynamic aspect of the framework allows caregivers to select assistive services needed for each patient, based on its cognitive health, in a timely manner.

5.2 Rational of our Deployment Approach

Our research in AAL aims at improving healthcare and quality of life for dementia patients during early stages of the disease when cognitive impairments are still mild and amenable to assisted intervention. We are interested in supporting healthcare in nursing homes by helping residents to perform their ADLs and providing support to caregivers. Therefore, we have focused on developing and deploying a technical solution that will provide assistive services to help residents and their caregivers.

Our approach consists of starting from a pre-deployment analysis conducted in a nursing home closely with end-users (dementia patients) and specialists in dementia care (professional caregivers) in order to identify the needs, develop and deploy a technical system based on the collected requirements, then evaluate the performance and usability of the proposed solution in real settings. The idea is that the developed system will not be only deployed for validation, however, it continue to evolve through the different phases of the project based on requirements and problems we identify during the deployment. Our approach has involved healthcare specialists in the design process, as recommended by Orpwood et al. [16]. Also, we have pushed further the idea to include professional people in the evaluation of the performance and usability of the proposed solutions in real life conditions.

We envision that such a multidisciplinary design approach, supporting a deployment in real life settings is crucial and that a simple system developed and validated in this way is more relevant and valuable than a well-featured solution proven stable only in a laboratory. In fact, most of the systems in the field of smart homes and dementia assistance work perfectly in a laboratory testing environment. However, they fall short when they progress toward commercialization or real deployment due to the lack of collaboration with professionals in the domain and the restriction of these studies to laboratories prototyping and testing.

Some ideas for building prototype environments [112, 113] are interesting, as they help to involve stakeholders in the design and testing process. However, real life scenarios are unlimited and cannot be enumerated and tested in prototyping environments where there are only a limited number of users and the technology is used only for a short time. In addition, these environments do not help to evaluate the technical usability of the designed system and the reaction of stakeholders in real world settings. Indeed, technical problems (sensors pulled off by patients, bad network connectivity, etc.) and design problems (household routines, multiple users, adaptability of the system to different patients' profiles, etc.) unpredictable in these environments during the design process and may only appear at the system delivery stage. These problems should be identified and resolved beforehand, during the development phase.

5.3 AMUPADH Project: A Top/Down Deployment Approach

The Activity Monitoring and UI Plasticity for supporting Ageing with mild Dementia at Home (AMUPADH) project is a two year project (2010-2012) between the Image & Pervasive Access Laboratory (IPAL, CNRS UMI 2955, France), the Institute for Infocomm Research (I^2R, Singapore), the School of Computing of the National University of Singapore (NUS, Singapore) and in close collaboration with Alexandra Hospital and Peacehaven Nursing Home in Singapore. It is one of the eleven A*STAR SERC Home2015 (Singapore national research program) projects, promoting a cross-disciplinary research enabling technologies, foundations or frameworks for the future home systems and technologies.

The AMUPADH project has focused on the automated recognition of activities and behaviors in smart homes and the provision of assistive services accordingly. This achievement is based on the automated monitoring of ADLs among single and multiple residents in nursing home.

The top/down approach used during the AMUPADH project consists of defining during design phase, the assistive services that will be provided for the patients, with no concern of technical constraints. Once the services are settled, required sensors are determined and installed. As an example, in order to detect a patient wandering during the night (which is a common problem for dementia patients), motion sensors are deployed on the nursing rooms' ceilings while pressure sensors are placed under the patients' mattress. Speakers are used to interact with patients.

5.4 Research Approach to Real Life Deployment

5.4.1 Choice of Deployment Environment

We have chosen to carry on our study first in a nursing home as it simplifies the recruitment of consenting residents and offers a semi-controlled environment, where professional caregivers are present on the field, permitting us to gather feedback and interest of stakeholders. We partnered with a nursing home hosting elderly patients with mild dementia who correspond to our targeted population.

Peacehaven, a nursing home from Singapore's Salvation Army, is our host and partner in the current study. It welcomes around 400 patients with dementia ranging from stage 4 to 5 according to the GDS. Residents on the second floor of this nursing home are at dementia stage 5 (moderate) while residents of the third floor have a mild dementia evaluated at stage 4. Each level has eight professional caregivers to assist residents, although residents from the second floor need more assistance and help. We have decided to conduct our study on residents with moderate dementia from the second floor where caregivers have a greater need of a solution to reduce their burden. The deployment has been planned in three phases preceded by a pre-deployment analysis and test phase as detailed in table 5.1.

Table 5.1: Timeline for the development and deployment of our solution in the nursing home (14 months trial)

Timeline	Description	Activities
Mar 2010 - Mar 2011	Observations, discussions & prototyping	Pre-deployment
Apr - June 2011	Prototyping & Demo	
Jul 2011	Application for ethics approval	
Aug 2011	Ethics approval obtained	
Aug - Oct 2011	Initial trial setup and field testing of system	
Oct - Jan 2012	First phase (1 room, 4-months trial)	Deployment + Ground truth + Data analysis
Jan - Feb 2012	Analysis, features update & performance tuning Initiate the work on the dynamic aspect of the framework	
Feb - May 2012	Second phase (1 room, 4-months trial)	
May - June 2012	Analysis & questionnaire survey	
June - Nov 2012	Third phase (3 rooms, 6-months trial)	
Nov - Dec 2012	Analysis & questionnaire survey Initiate the work on uncertainty handling	

5.4.2 Pre-deployment Observations and Discussions

Before starting the deployment, we have decided to conduct weekly-observation sessions with residents from the nursing home during a period of three months and to organize focused group discussions with caregivers. As residents have a daily-common schedule, observation sessions consisted of following the daily schedule with two or three of them and participating in weekly group activities (organized with 10 to 15 residents). Table 5.2 illustrates some of our observations.

Table 5.2: A snapshot of data collected from observation sessions

Type of observation	**Observations**
Rooms observation	• Two or three beds in each room • One bathroom is attached to each room • Food and medications are taken in the common area • TV is only available in the common area
ADLs observation	• One resident (level 2) keeps on washing hands or showering for hours • Assisting shower is very difficult for nurses • Patients forget to turn off the taps • Patients need instructions of what to do next • Patients need encouragement to initiate activities • They forget to continue activities if they were interrupted, e.g. forget to finish eating after going to the toilet • They forget things they have already done, e.g. some may shower too often
Group Activities observation	• 10 to 15 residents • 2 to 4 professional caregivers • Participants are divided into small groups • Caregivers need to give instructions

These observations have been followed by discussions with caregivers in order to gain a comprehensive understanding of the living conditions at the nursing home. Focus group discussions have been organized each time with around 5 professional caregivers and doctors to discuss about the collected observations, present some demos and discuss about possible improvements, as well as patients' and caregivers' requirements, needs and issues.

Discussions are tape-recorded and have started with questions collected from the field and previous meetings. Then a session dedicated for updates, exchange and interaction last for around 2 hours. After that, collected information are analyzed and processed to produce meeting reports. It was difficult to extract meaningful information from discussions with patients due to their dementia. For example, they may speak about events that happened a long time ago as if they were happening during the current day.

5.4.3 Participants' Characteristics and Selection Process

The staff from the nursing home selected potential subjects, in consultation with the doctors, and the appropriate residents are approached to participate in this study. Ill patients with unstable parameters or life-limiting diseases (such as cancer or end stage heart failure) have been excluded from the study. Those with pacemakers and other required medical electronic devices for monitoring or treatment (monitors like telemetry, ECG, pulse oximetry, infusion pumps, etc.) have also been excluded. We have selected only patients who could give informed consent in spite of dementia, based on a mental competence assessment of the patient done by a clinician steeped in dementia care, or patients with dementia who have had a Legally-Appointed Representative (LAR, usually a close relative) who could provide informed consent on their behalf if they lack mental capacity to do so. The nursing home staff selected to take part in the study has been those who are actively-involved in the day-to-day care of the selected residents and know quite well their habits.

Two patients living in the same room and two caregivers have first been involved in the study. Later on, six other patients from two other rooms (3 in each room) have granted approval to participate to the trial. The eight patients are women living on the second floor of the nursing home and aged between 78 and 92 years old. Two of them need minimal assistance (help to walk or to lie on the bed) while the six others need moderate assistance (help to take the shower or to eat). Table 5.3 gives an overview of the different patients' profiles.

To proceed with the deployment in real settings, the study has been ethically approved by the Institutional Review Board (IRB) of the National University of Singapore (NUS) under the number 11-222. After the *Consent Forms* were signed and collected, and before the start of the deployment, more visits have been organized to analyze personalized needs of selected patients with recourse to professional caregivers implicated in the study. Several

Table 5.3: Participants profiles

Patient	Age	**Functional Status**	**Room nb**
Patient 1	90	Needs minimal assistance*	8
Patient 2	92	Needs moderate assistance**	8
Patient 3	85	Needs moderate assistance	8
Patient 4	79	Needs minimal assistance	9
Patient 5	87	Needs moderate assistance	9
Patient 6	92	Needs moderate assistance	11
Patient 7	82	Needs moderate assistance	11
Patient 8	78	Needs moderate assistance	11

* Minimal assistance: provided assistance consists only on elementary activities (walking, lying, etc.)
** Moderate assistance: provided assistance includes more critical activities (eating, toileting, etc.)

assistive services related to each patient's problems are identified.

5.4.4 Deployed System

The deployed system is composed of low-cost and non-intrusive sensors (e.g. pressure sensors, proximity sensors, vibration sensors, motion sensors), different devices of interaction (e.g. speakers and tablets for the residents, smart-phones and a nursing console for the caregivers) and a centralized compact machine deployed in each room. Sensors are used to monitor the residents and acquire low-level context information while interaction devices are used to provide reminders and notifications for patients or caregivers.

Figure 5.1 shows some types of sensors used in our deployment. RFID reader in "A" is used for residents identification. It was tested with RFID tags worn by residents as wristbands to identify the patient using the bed or going inside the toilet. Pressure sensors in "B" are used to detect the resident lying or setting on the bed. In "C" we have motion sensor used to detect movement. It was used to detect residents' presence in the bedroom, toilet or wash-room. Finally, vibrator sensors in "D" are used to detect the use of tap while washing hands or showering.

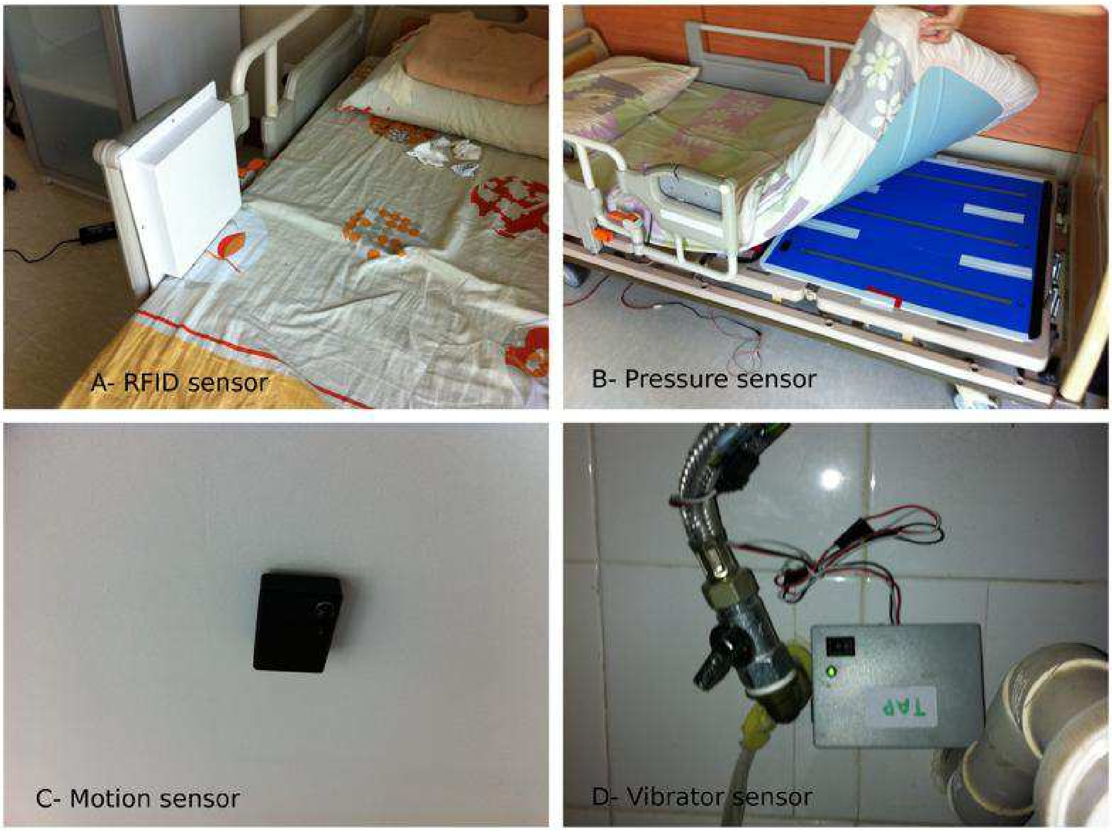


Figure 5.1: Sensors deployed in the residents' room

Deployed devices are presented in Figure 5.2. Bluetooth speakers in "A" and "B" are used to send reminders to the patients in their bedrooms or bathrooms. The communication with the caregivers is achieved using smart-phones in "C" or the nursing console in "D" through alerts and notifications.

The nursing home environment with the different sensors and interaction devices is described in Figure 5.3a while Figure 5.3b shows the hardware composition and communication. In each room is installed a compact fanless Debian machine (115 x 115 x 35 mm, 505g), mounted with a 500MB RAM/500Hz CPU, a 8GB Compact Flash drive, the whole consuming only 5W. Sensors are using the ZigBee communication protocol on a wireless sensor network based on Crossbow's IRIS mote platform. A Crossbow node is connected via serial port to the Debian machine, serving as gateway. The communication with other devices in the environment uses Bluetooth for residents' embedded speakers, a client-server communication over Wi-Fi for the nursing console (Windows 7 machine with touchscreen) or 3G for the nurses' smartphones (Samsung Galaxy S2 with Android 2.3

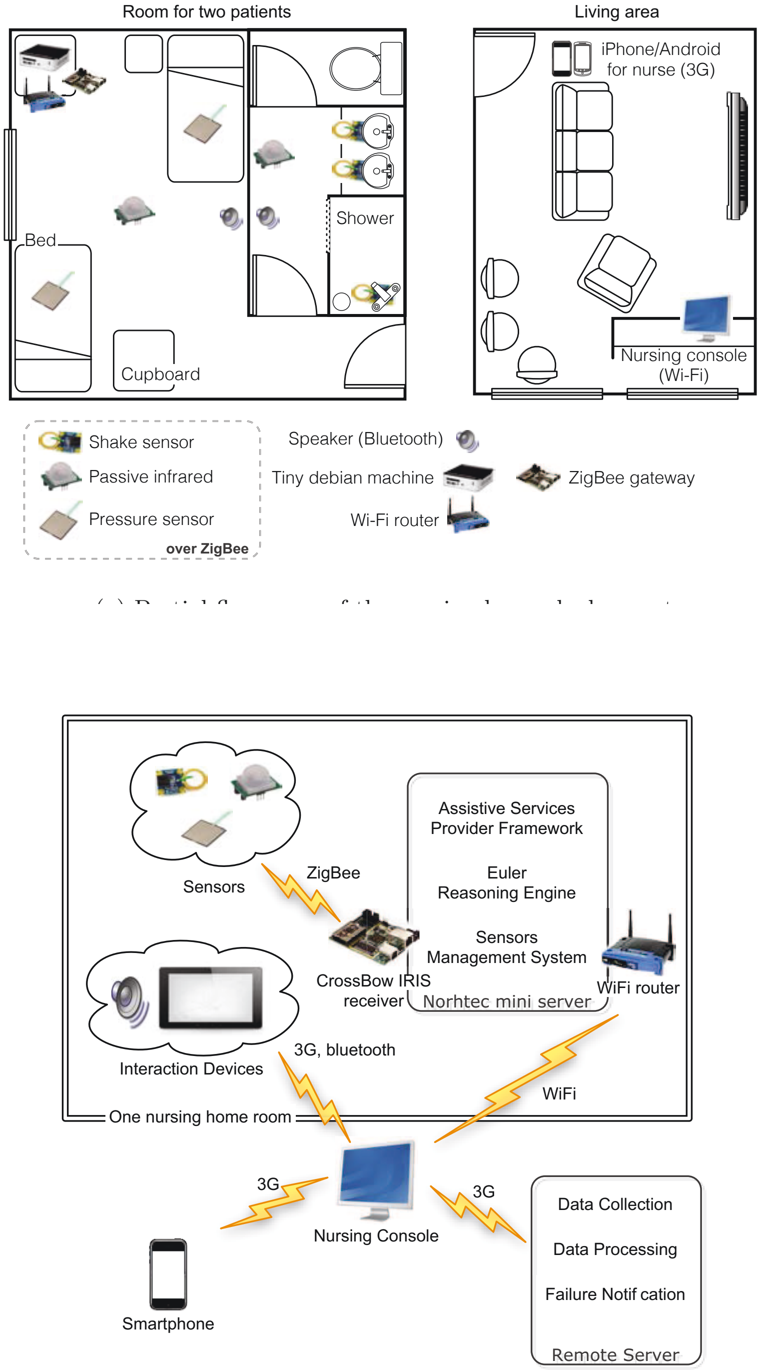
Figure 5.2: Devices deployed in the nursing home

and Apple iPhone 4 with iOS 5) and tablet PCs. The Nursing console also serve to receive collected data from the Debian machine then forward them through 3G to a remote server in our premises. The remote server is responsible for data collection and processing and also for hardware failure notifications.

Within the system, our UbiSMART framework, assuring the context aware reasoning and the provision of assistive services for the patients, has been developed based on human and technical requirements gathered from the pre-deployment analysis and installed on the Debian compact machine of each room. On the Debian machine are running the assistive services provider framework, the Euler reasoning engine and the sensors management system.

5.4.5 Data Gathering & Performance Evaluation

In order to evaluate the performance of our system, and as we have committed ourselves not to use video recording to preserve the privacy of the nursing home's residents and



(b) System hardware deployment

Figure 5.3: System deployment in the nursing home

caregivers, we have chosen to rely on log-sheets filled by the caregivers and compared with our system logs. We have collected all system log data during our trial period of 14 months and have extracted meaningful information concerning sensors states and patients' context, then we have compared them with the caregivers' log-sheets. Log-sheets provide the ground-truth: caregivers have been asked to fill them in with hourly information about patients' location, patients' abnormal behaviors and possible remarks. Table 5.4 is a sample of blank caregivers' log-sheet which has been used for our ground-truth data collection through data entry into a database.

Table 5.4: Sample of caregivers' log-sheet

Date: 19/02/2012 Patient: Patient 1

	Where Is Patient Right Now?	**Did Patient Shower For Too Long?**	**Did Patient Forget to Turn Off Tap?**	**Did Patient Forget to Flush Toilet?**	**Did Patient Wander Around Aimlessly?**	**Did Patient Ask For Something?** If Yes What Did (S)he Ask for?
10:00:00	Bathroom	No	Yes	No	No	No
11:00:00	Common Area	No	No	No	No	No
...						

Next, I present the outcomes of our pre-deployment analysis and the different human requirements raised. I also provide the evaluation of our proposed system performance and I present the results of our data analysis.

5.5 Outcomes of the Pre-deployment Analysis

In addition to the technical requirement mentioned in chapter 1, on which we have based our research approach for building a dynamic service-oriented framework handling uncertainty, different other human requirements have emerged from our pre-deployment observations.

From our pre-deployment observations and discussions we have gained an understanding of the living conditions at the nursing home and the different problems that residents and caregivers are facing. We have realized that although residents are free to move around in the common areas –in fact, they do so quite frequently– The most of the unsupervised time has been spent in their bedrooms or in the washrooms. Hence the bedroom and its attached washroom are selected as the main areas for our study.

Caregivers Relief In our observations, we realized that most of the assistive tasks performed by caregivers consist of encouraging patients to start some activities, showing them instructions about the first several steps to follow, serving medication or asking patients to drink water. It is usually difficult to follow and assist all the residents' activities, especially during the night when there are fewer caregivers on duty and patients are prone to more critical situations. Therefore, caregivers have considered that reminders and notifications could be helpful for them to be informed on how patients are performing their ADLs, and to provide support as and when appropriate. In order not to increase the attention needed from caregivers, it has been decided that the system should remain on standby when no issues are met by patients. Notifications are to be raised only when a reaction is necessary.

Independent Ageing Residents are having problems such as showering repeatedly (some may shower 3 times in the morning because they forget the activities they have already done), remain in the shower for too long (they stand in the shower and let the water running for an extended period of time since they do not know what to do), wandering (some just walk around in their rooms spending a sleepless night) or using other residents' belongings (sleep on the wrong bed or wear someone else's clothes). In order to help these patients overcome their problems, caregivers have emphasized that reminders should first be sent to patients to encourage them to think and retain some level of independence. Caregivers should interfere only when the residents lose their way and are not able to solve their problems independently.

Level of Dementia Due to our dedicated design approach and validation environment, and in order to make sure patients experience an optimal engagement and usefulness, a requirement rises on the most suitable level of dementia for assisted patients. Indeed, as introduced earlier, elders who are still able to live purposefully without assistance would not feel the need for such system, hence acceptance would be rather low. On the contrary, the system is designed to help patients keep a certain level of autonomy and therefore rely on their understanding of reminders, e.g. based on their capacity to recognize the purpose of some objects in graphical reminders. This means that patients with a pronounced dementia, in this case an advanced agnosia, would not be able to react to reminders, which would only build up their frustration. From our exchange with clinicians, it has been determined that patients with dementia at GDS level 5 would experience the most suitable assistance.

Personalized assistance Each patient has his own habits and caregivers should assist them in an adapted manner and with different activities. Using our system, caregivers should be able to register an interest that keeps track of which services are useful for which residents. Services should also be personalized to the profile and the evolution of the disease of each patient.

Based on these outcomes, we have fixed a set of possible assistive services that can be provided for the nursing home (using a wrong bed, taking a shower for too long, taking a shower without using soap, wandering, leaving water running, falling detection, spending sleepless night), mainly via reminders or notifications. These reminders should be completely based on non-intrusive technologies, especially without any resort to video analysis. We have decided to focus on some key assistive services that are interesting for the caregivers and that can be realized by our team to be initially demonstrated in the laboratory. Four services (using wrong bed, showering for too long, showering without soap, not sleeping at night) have been selected. The nursing home staff has appreciated the lab demo and has provided positive feedback. The demo has been then showcased in real conditions within the nursing home, without including real end-users (caregivers and patients). Feedbacks mainly focused on the reconfigurability, customization and adaptability of the system which were the starting points for further development of our system.

5.6 Deployed Services

Our solution was deployed in 3 rooms within the pilot site nursing home and includes 8 patients for the trial. The deployment has started with an initial trial set-up and field testing of the system for technical validation, without any interaction with the patients and caregivers. Then three phases have been planned; during the first two phases, the interaction was only with the caregivers to collect feedbacks and analyze the performance of the system. Patients have not been included in the interaction to test the system without affecting residents with eventual false alarms. The interaction with patients was initiated during the last phase when we have significantly reduced false alarms generated by the system and increased the system uptime.

We have provided 4 assistive services which have been identified with the help of the caregivers from the nursing home based on some of the most frequent issues that participating residents have been experiencing. We can easily define specific problems for the residents and provide tailor-made assistive services using the dynamic aspect of our software platform. The Four assistive services provided are: 1) wandering at night, 2) showering for too long, 3) leaving the wash-room tap on and 4) toilet fall detection, which not only correspond to actual needs for our residents but also represent dangerous situations when they are in dark or wet environment, and increases the likelihood of a fall. Table 5.5 gives an overview of these 4 services:

Wandering at night Some of our residents usually wake up during the night, go to the toilet, then start wandering in the room without going back to sleep. We have deployed a service to detect this abnormal behavior. Pressure sensors are deployed under the mattresses of the patients' beds to detect their presence. During the night, if the patient is not detected on his bed and the motion sensor placed on the bedroom's ceiling is firing, the system infers that the patient is wandering and a notification is sent to the caregivers on their smart-phones and nursing console. There are no interactions with the targeted patient to avoid disturbing the co-resident during the night.

Toilet fall detection Falls represent critical situations for the patients and need prompt intervention from the caregivers. They have been found to be more frequent and crucial in the toilet. In order to quickly detect these situations and alert the caregivers, we have deployed motion sensors in the toilet to detect the presence of the patient, as well as attached proximity sensors to the ceiling to measure the height difference of the patient. Once we detect a sudden change in the height, and after a while, if the patient is still in the toilet and the height has not changed, an alert is sent to the caregivers on their smart-phones and nursing console.

Showering too long As it has been previously explained, during the shower, some residents stand in the shower and let water run for an extended period of time because they do not know what to do. This can be detected by the system, which alerts the patient

Table 5.5: Assistive services deployed in the nursing home

Assistive services	Sensors involved	Interaction devices
Wandering at night patient starts wandering during the night without going back to sleep	**motion sensor** something moving inside the room **mattress sensor** patient is not in his bed	**smart-phones** **nursing console** notification for caregivers to help the resident
Toilet fall detection patient detected as lying on the toilet's floor	**motion sensor** patient detected in the toilet **proximity sensor** calculate the height difference between the patient and the ceiling	**smart-phones** **nursing console** alerts for caregivers to help the resident
Showering too long patient using the shower for a long period of time	**motion sensor** patient detected in the shower **vibrator** attached to the pipeline to detect the use of shower	**speakers** ask the patient to end the shower **smart-phones** **nursing console** notification to caregivers if problem not solved
Leaving the wash-room tap on Patient left the tap on after washing hands	**proximity sensor** patient detect near the sink **vibrator** attached to the pipeline to detect the use of the tap	**speakers** ask patient to turn off the tap **smart-phones** **nursing console** notification to caregivers if problem not solved

or sends a notification to the caregivers. To ensure this, a motion sensor deployed in the shower detects the presence of the patient while the vibrator attached to the water pipeline detects that he has started showering. Another complementary assistive service *shower without soap* has been suggested during the pre-deployment period, to detect that patient is not using the soap during the shower and to notify him to do so. However, caregivers were interested only in the *showering too long* assistive service for the patients involved.

Leaving the wash-room tap on Some patients forget to turn off the tap after washing their hands, which may lead to an overflow of water in the washroom and patients may slip and fall down. In order to prevent this from happening, we have deployed a service to detect when the tap is left on. The combination of the proximity sensor to detect the patient near the sink and the vibrator on the water pipeline to detect that the tap is on, infers that the patient is using the tap. Once the vibrator is still firing and the patient is no more near the sink anymore, a vocal message is sent to the patient through the speakers to remind him to turn off the tap. The reminder is repeated for a fixed number of times configured by the caregivers and personalized for each patient. If the resident ignores the reminders, a notification is sent to the caregivers.

Some research works have provided taps that are automatically turned off in case of flooding [16]. This could easily be integrated in our platform but it is not our objective here. Indeed, in our case, patients should be motivated to perform their activities in the right way, and caregivers should be informed about the evolution of their patients' disease and how well they are performing their activities. For example, as shown in Figure 5.4, we have noticed one of our patients' situation deterioration. Normally she has less than 6 reminders each day, but suddenly, we have detected an increase of this number up to 12 reminders. This would suggest that the patient's situation has worsened around 4^{th} of March which has been confirmed by the caregivers as they have also realized that she was having serious problems soon after this period.

In addition to these services, we have developed a *next hour activities prediction service*, based on the data analysis of log sheets from caregivers and system logs. This service allows the prediction of patients' activities (bathroom, common area, and room activities) in the next hour using decision-tree ensemble and active learning with various parameters. This service was tested with patients 4 and 5 (Room 9) from March to October 2012 (8 months). We have reached an accuracy of 91% and 68% respectively for sleep and toileting

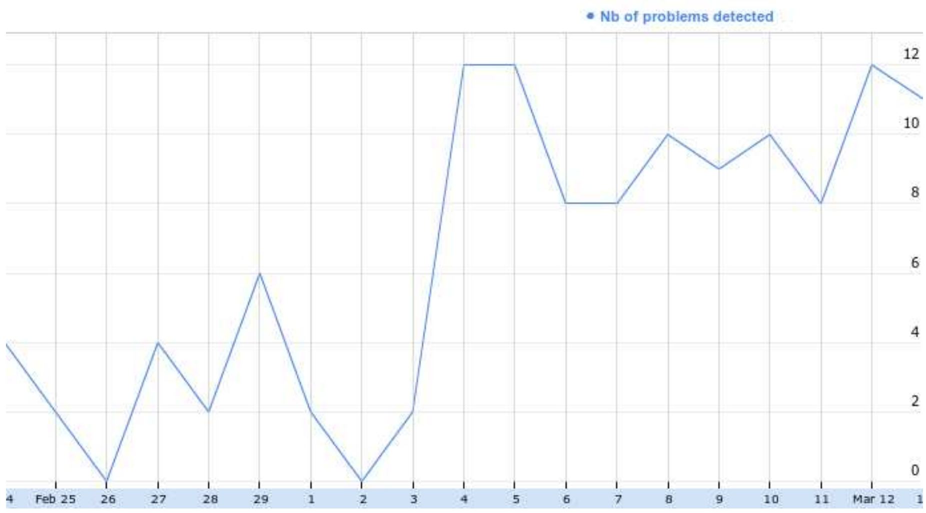


Figure 5.4: Evolution of patient's detected problems

activities prediction. This service allows the improvement of caregivers' planning and time management especially during the night as they can predict the patients' activities an hour in advance and assist when necessary. The whole system gives them more time to care for more patients.

5.7 System Performance

Based on the analysis of our system data log, we have classified *atomic* events – e.g. the use of tap, shower and locations – which happen 34 times a day in average and *complex* events – corresponding to deviances and services provision – which happen 7 times a day in average. The comparison between these results and the ground-truth's log-sheets during the first phase of the deployment reveals a matching rate of approximately 71% on a weekly basis analysis.

The 29% of false alarms detected are related to distinct reasons. They could emerge from human errors caused by caregivers imprecision in the logging time and by the fact that other patients are able to use the wash-room used for our deployment. At the same time, it is also related to technological failures such as sensors failure (59%) (sensor removed, sensor out of battery, sensor packets lost), reasoning failure (12%) (software bug) and connection problems (29%) (Wi-Fi disconnection, ZigBee communication problem). Our deployment at the nursing home allowed us to highlight these main areas in which the system needs to be improved. Figure 5.5 shows the relative occurrence of each of these

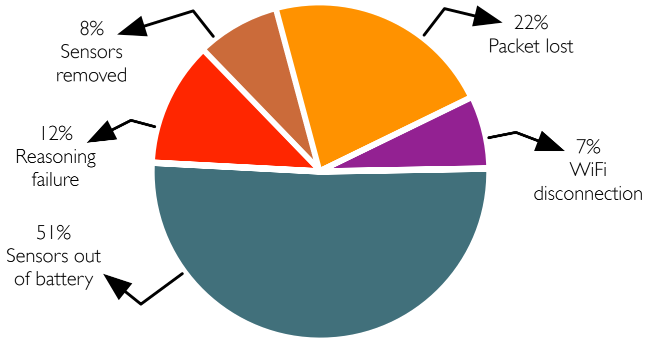
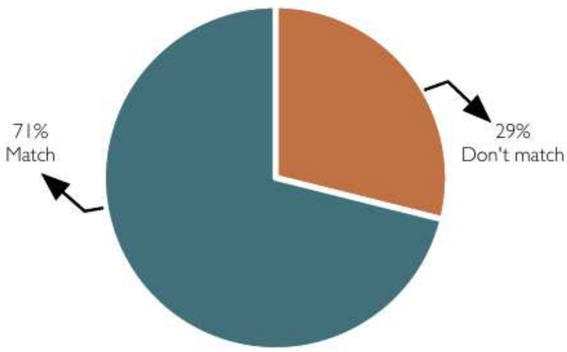
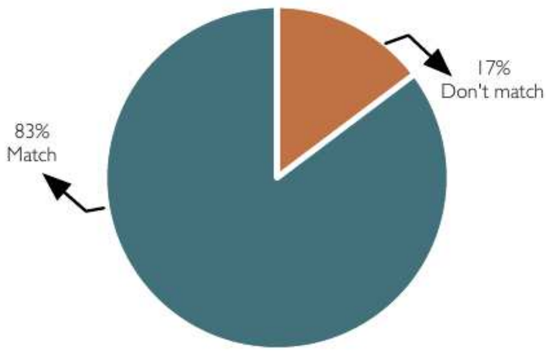


Figure 5.5: Pie chart for system crash reasons in the nursing home

We worked hard on improving these aspects and were able to improve the average uptime of the system from 3 days in December 2011 to 11 days in May 2012. We also reached an 83% matching rate in the weekly context recognition performance during the last phase of deployment. Figure 5.6 shows the evolution of system performance from the beginning to the end of the deployment.



(a) February 2012



(b) September 2012

Figure 5.6: Weekly based context recognition performance evolution (in the nursing home)

The more technical errors have been considerably reduced, notably the batteries failure, with the use of a new type of batteries and the development of a protocol which involves acknowledge messages to switch off the sensors' radio usage when it is not used. This has increased the battery lifetime from 5 days to 15 days. We have also worked on improving the reasoning engine performance through formal verification of defined rules using model checking techniques. This makes us detect some unwanted situations such as non-reachable rules which do not affect the system precision but increase the complexity and reasoning time, redundant rules causing multiple decisions about the same situation and logically conflicting rules which can lead to conflicting and non-logical situations [114]. The technical problems encountered during this deployment have been the basis to initiate our research on uncertainty handling. Incomplete information derived from faulty hardware, delays between production and consumption of the information, or even networking problems, leads the work to concentrate on the incorporation of the uncertainty notion derived for hardware characteristics in the reasoning process.

Another aspect we have considered as estimation of the performance of our system is the time needed to set it up into a new environment. Before I implemented the dynamic approach and as we had the context-awareness part integrated in an imperative manner, it took 3 days for a team of 2 research-engineers to install and adapt the system in the nursing home environment. However, moving to the dynamic approach and the semantic modeling, adaptation to a new environment with its specificities took us only 2 to 3 hours, most of which was to adapt the semantic model. This has been reduced as the system intrinsic logic is kept unchanged. Once the system is installed, we computed its reaction time; calculated between the time a service is needed and the time it is delivered. This has an average of 2.7s, which has been refined into 1.2s for reasoning process, 0.7s for communication between modules and 0.8s for the processing due to other miscellaneous bundles. The effective time for integrating a new assistive service into the platform was about 2s for the deployment of one assistive service with two sensors and one interaction device.

5.8 Data Analysis

From two data sources (caregivers' log-sheets and system log), we provide two data visualizations of a resident's behavior which illustrate that we can provide an early detection of serious problems. This is highlighted through one of our patient's (Patient 5) data analysis. The nursing home staff informed us that this patient was seriously ill and had to leave the nursing home for medical treatment in late-March 2012, which was detectable in the data gathered and corresponding to the patient's behavior in early-March. The proposed visualizations expose the patient's relative density of activity in bathroom (e.g. pass urine or motion), common area (e.g. watching television or eating), or bedroom (e.g. sleep) when she starts to have medical problems and demonstrate the possibility to recognize the deterioration of her situation.

Figure 5.7 is a chronological heatmap of caregivers' log-sheets (ground truth) for patients 4 & 5 in February and early-March 2012 (1.5 month). The comparison of the two patients' activities demonstrates that patient 5 has significantly more bathroom activities and less common area activities (number of orange blocks) and that she spends more days away from nursing home (the gray blocks between colored ones) for hospital treatments. Focusing on patient 5's data across the investigated period, we notice a significant increase in bathroom activities in March 2012 compared to February with more orange blocks. It is very likely that this patient had serious medical problems during this period with more frequent bathroom visits, less tendency for leisure activities and more hospital visits for medical treatment.

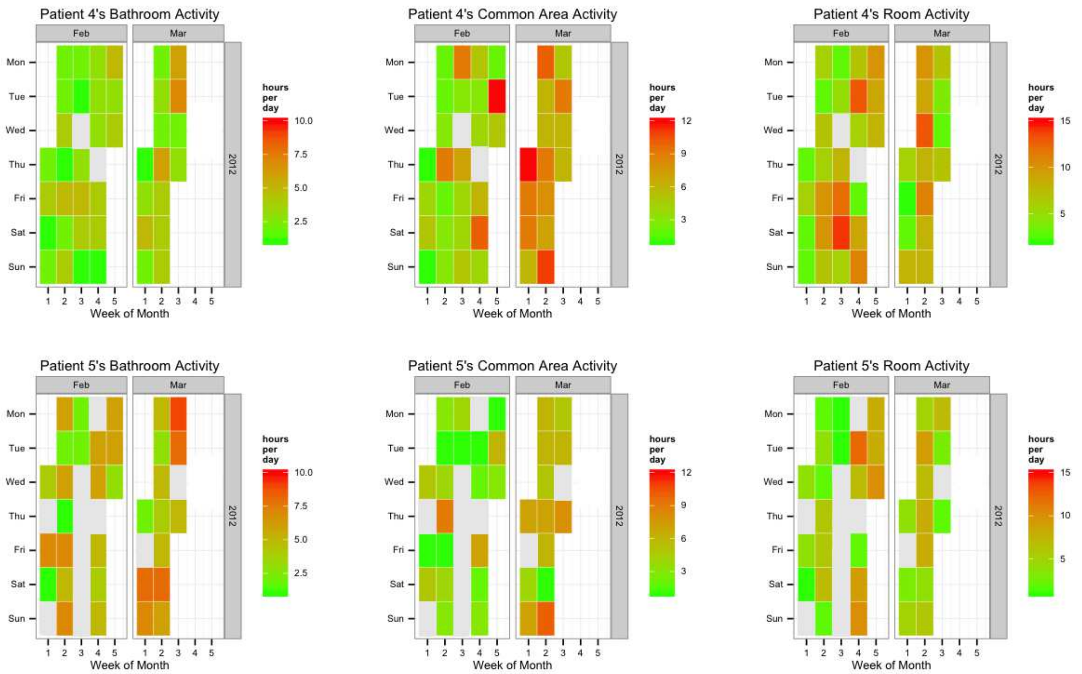


Figure 5.7: Chronological heatmap of caregivers' log-sheets (ground truth) for Patients 4 & 5 (Room 9) in February and early-March 2012 (1.5 months)

This is also confirmed by Figure 5.8, which represents a chronological heatmap of system log data (sensor data) cross-referenced with caregivers' log-sheets (ground truth) for the bathroom activity of the two patients in early-March 2012 (7 days). Visits to toilet are characterized by high amounts of vibration and motion sensor firings while sleep is related to pressure sensor firings. When comparing Patient 4 with Patient 5 for 7 days, Patient 5 has almost twice as many bathroom activities as a normal resident.

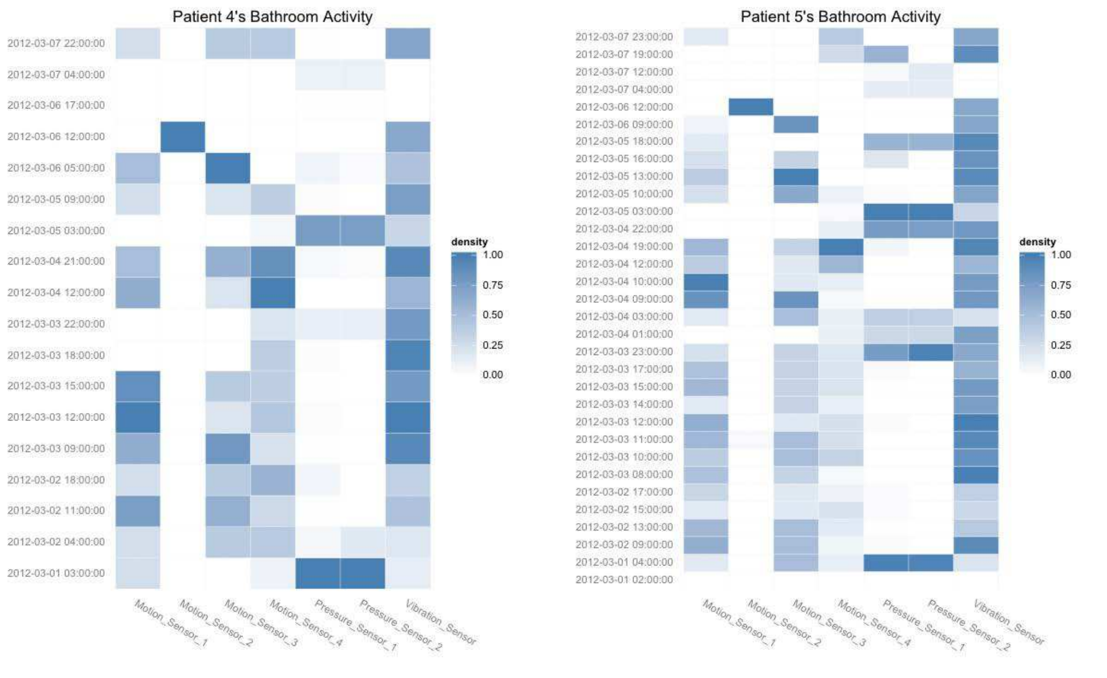


Figure 5.8: Chronological heatmap of system log data (sensor data) cross-referenced with caregivers' log-sheets (ground truth) for Patients 4 & 5 (Room 9) in early-March 2012 (7 days) for bathroom activity

5.9 Qualitative Feedback

The feedback received from the Nursing Home staff has been very encouraging. The deployment was promising in terms of demonstrated features and capabilities. In view of the expected benefits, the Head of the Nursing Home has remarked that the staff would like to have the full system deployed in every room. This was encouraging since there has been a perceptible change in attitude over the months during which the deployment had taken place. The caregivers-staff have become more adept in the use of smart-phones and have appreciated the value of the underlying sensor based technology. Though the staff complied with the completion of the manual log sheets, which was crucial as it was the only form of ground truth available in the absence of video camera based logs, they admitted having difficulties to cope systematically with the extra work it represented. We discussed simplifying the logging process by providing a more automated logging media through tablet PCs embedded in the environment and bringing logging down to a few clicks on touch-screens. Doctors carry a positive attitude towards the deployment and feel that it would go a long way to improve residents' safety and would add to the well-being and comfort of the residents.

5.10 Discussion & Lessons Learned

From a human perspective, requirements consist mainly of providing adapted assistance with different activities based on each patient's habits and profile. This assistance encourages the patient to start/stop some activities or show instructions on the first several steps of each activity. Caregivers need this kind of assistance mostly in bedrooms where patients spend the majority of their days and during the night when there are less caregivers. Patients whose dementia should be at GDS level 5 for an optimal experience need to perform their activities in the right way and caregivers should be informed how well they are doing. Therefore, the provided assistance should incite patients to think and retain some level of independence in performing their activities without just solving problems. Caregivers should interfere only when the patient cannot resolve it.

From a technical perspective, the system should guarantee the privacy of patients and caregivers, and manage a group of people in controlled areas. It should also be reconfigurable, customizable and adaptable allowing caregivers to select different assistive services for selected residents and adapt them to patients' profiles and diseases' evolution. Caregivers consider that reminders and notifications are helpful to provide assistance to patients and inform them about their situations. In order to cope with technical problems resulting from faulty hardware or network problems, the platform should be designed for failure that will allow system recovery, crash analysis and crash reports.

The real life deployment approach adopted does not only allow us to fix these different requirements but also proved to be a source of different lessons learnt. In fact, different challenges have arisen from our deployment in real settings. Problems such as batteries life, sensors removal or network connection cannot be identified while performing experiences in the laboratories under optimum conditions. Deployment has allowed us to expose the reasoning engine to more realistic and complex scenarios, thus leading to more robust and concrete performance evaluations. In addition, the involvement of professional caregivers in the design and evaluation process was very helpful, in making sure we provide a system that could respond to stakeholders' and end-users' needs and helps with the awareness and acceptability of such systems.

Based on the ability of the platform to detect specific behaviors and using implicit information contained in sensor data as well, we hope we will provide services that would help detect long-term shifts in the habits of patients as this represents a meaningful clinical element to provide doctors and/or family members with. It has been shown in the data analysis section that some hindsights are already available as we can predict a few days in advance the degradation of a patient's condition. At first, we have considered gathering manual log-sheets filled by nurses on the field to get some approximated ground-truth that would help us evaluate the system's recognition performance. However, as these log-sheets were as spare as our sensor data, it has been very difficult crossing the data to estimate the performance. The interesting point though, is that we unexpectedly have realized that data from manual nurses' log sheets was at least as valuable as our sensor data, in the sense that by using the same kind of algorithms we were able to extract very meaningful information about patients' habits and health. One of our major lessons learned is that providing a more automated, electronic and pervasive manner to gather this data from nurses on the field would be a very useful feature to integrate into further AAL solution; naturally with backend algorithms analyzing this data and making valuable clinical knowledge available.

The adopted deployment approach requires a lot of time and manpower which may cause deliverables delays. In fact, we have been working on solving technical problems and adopting new approaches that are required for real life deployments in order to increase the system accuracy form 71% to 83% and uptime from 3 to 11 days. In addition, the use of the modular and declarative approaches is beneficial to guarantee the dynamism and adaptability of the system but requires specific skills with a time-consuming learning curve. In order to increase the system uptime, sensor batteries lifetime has been increased from 5 to 15 days, which is still considered to be short during the deployment. We are trying out a different sensor platform and we have reached 4 months of autonomy so far.

Despite our deployment of interactive devices for the residents and the extension of the trial from two to eight patients, the number of participants was limited, hence a significant amount of data is missing to provide meaningful results about the impact of the system. Further surveys concerning the impact on social and health aspects are needed, and should be realized during future larger scale deployments as now we have a system that is quite stable and quickly deployable and adaptable to different environments.

5.11 Conclusion

In this chapter, I have presented the adopted approach to deploy an ambient assistive living system with the UbiSMART framework for patients with dementia. This approach is based on a pre-deployment analysis with implication of professional caregivers to set different human and technical requirements that the system should fulfill. The deployment was a source of inspiration which lead our research work to focus on the dynamism and uncertainty handling in our UbiSMART framework. It has also allowed us to make performance tests in real conditions and to get feedback from the stakeholders.

6 Conclusion

During this thesis, we have been working on realizing an ambient assistive solution for the ageing population to help them keep some level of independence and maintain a convenient life. Our interest in the elderly population emerges from the study presented in the first chapter showing the diverse impacts of ageing on the person's lifestyle and the alarming increase of the number of dependent seniors and their care cost. It has been demonstrated that this situation is more related to the growth of the ageing population and the decline of the fertility rate worldwide. From this study was deduced the need to develop ambient assistive systems and assistive technologies to compensate the decline of ageing people cognitive and physical capacities.

Even though different ambient assistive systems have been developed to respond to this need, most of them have not been deployed which inhibits them from meeting the true requirements of the real conditions and drive them apart from the technical challenges of the field settings. The approach we have adopted was to perform our research in parallel with a real-world deployment. We have collaborated with a nursing home in Singapore which was used as a support to conduct our investigations and discussions during a pre-deployment period then as a deployment site to evaluate the performance of our proposed solution. This deployment was carried out in the context of the AMUPADH project, an A*STAR SERC Home2015 (Singapore national research program) two-years project (2010-2012) between the Image an Pervasive Access Laboratory (IPAL, CNRS UMI 2955, France) working on the development of the framework, the Institute for Infocomm Research (I^2R, Singapore) focusing on the development and deployment of sensors and the School of Computing of the National University of Singapore (NUS, Singapore) interested in formal methods and rules verification.

Our proposed framework has evolved during this project period from a static and monolithic architecture with imperative reasoning to a dynamic and extendible framework, using the modular and declarative approaches which handle the uncertainty in the context information. In fact, our on-site investigations and discussions with professional caregivers and doctors from the nursing home allowed to identify human and technical requirements that need to be tackled in order to meet the patients' exigencies and for the system to be functional in real conditions. The main technical requirements on which I have focused in this thesis are, on one hand, the dynamism and extendibility of the framework and, on the other hand, the management of uncertainty in the context-aware reasoning. The dynamism ensures the adaptation of the framework to different deployment environments and heterogeneous patients' profiles and needs, while the uncertainty handling copes with the multiple hardware failure and network problems.

The dynamism of the framework was achieved using the modular and declarative approaches. It encompasses three main layers: the service model layer, the services interoperability layer and the reasoning layer. First, on the service model layer, the Open Service Gateway initiative (OSGi) was adopted to implement our modular approach. It allows representing system's entities (assistive services, sensors and interaction devices) as modules interacting with each others and has the possibility to communicate with entities represented as web services. I have explained our choice of the OSGi specification and of the Apache Felix implementation of OSGi used in the development of the UbiSMART framework. Second, on the services interoperability layer, I have shifted from the use of ActiveMQ which provides basic communication capabilities based on non-standardized defined XML schemes, to the use of DPWS which ensures a mechanical Plug&Play mechanism based on standardized communication techniques. I have provided our design of the UbiSMART framework with the different modules developed. Modules are deployed onto three different OSGi containers: the OSGi sensors gateway container, the OSGi devices gateway container and the OSGi assistive services container. The DPWS mechanism ensures the communication between the sensors and devices containers on one hand, and the assistive services container on the other hand. Finally, on the reasoning layer, I have adopted the declarative approach which ensures the separation of the application logic from the underlying models describing the peculiarities of the environment. I chose to use the semantic web technology to implement the declarative approach. I have provided a semantic model that describes the AAL environment and a set of first order logic rules to infer patient's context and provide assistive services. Based on these different techniques, I have proposed my semantic Plug&Play approach allowing the integration of new assistive services, sensors and interaction devices to the framework at runtime, then their integration into the reasoning process so that the new assistive services could be provided for the appropriate users.

In order to cope with uncertain information in AAL environments emerging from technical problems such as sensor hardware failure or network problems, I have proposed an approach for uncertainty measurement, representation and reasoning. This approach implies making some changes to the three layers presented above. First, on the service model layer, new modules have been integrated in order to perform the uncertainty measurement and decision making under uncertainty. I have provided a method for the measurement of information's confidence based on sensors' characteristic and functional parameters. In addition, I have used an implementation of the DST as a support for our reasoning engine to take decision about conflicting situations emerging from technical problems. Second, on the services interoperability layer, I have integrated the transfer of sensors characteristic parameter and information uncertainty measurement in the communication between the different modules. Finally, on the reasoning layer, I have provided a model which allows the semantic representation of entities' relations with the notion of uncertainty and integrated it in the semantic first order logic rules. The proposed model allows to keep the classical semantic representation and to make changes only for specific parts when the notion of uncertainty is needed. The changes in semantic rules elevate the uncertainty from sensors' raw information to higher-context data allowing their use for conflicting situations reasoning.

In summary, the main achievements of this thesis consist in:

- An ambient assistive living system for elderly people with age-related diseases.
- A notion of dynamism allowing the integration of specific assistive services for each group of patients.
- A semantic Plug&Play mechanism allowing the integration of new assistive services, sensing technologies and interaction devices at runtime, then incorporate them in the reasoning process.
- A semantic model for environment description allowing the separation of the application logic from the underlying context representation.
- Semantic reasoning rules for assistive service selection and provision
- An approach for uncertainty handling in AAL environments
- A method for uncertainty measurement based on sensors' characteristics and functional behavior
- A semantic model for uncertainty representation and reasoning.
- A combination of semantic with DST for data fusion and doubt lifting in conflicting situations.

The UbiSMART framework with the dynamic aspect was deployed in the nursing home in Singapore. A performance validation was realized through data analysis of collected data and a positive qualitative feedback has been obtained. However, some research work still need to be improved. For the semantic Plug&Play approach, I need to improve the rules generation part in order to handle more complex situations and provide high-level rules. Rules composition based on generated rules is another aspect that needs to be handled. We can also consider rules auto-generation after some basic bindings. On the uncertainty handling part, I may work on extending the proposed approach by considering new parameters and including the human behavior in uncertainty measurement and decision making. In addition, the approach used was not validated in real setting even though we have used off-line data collected from the new project *"Quality of Life" (QoL)* to demonstrate the results provided. I consider deepening the research on uncertainty handling within the two new projects *"Quality of Life project"* and *"VHP Inter@ctive project"*. Context reasoning under uncertainty could be extended to context prediction. In fact, reasoning about current situations is not always relevant for the end-user. This implies the need to predict context situation especially in cases of emergency and security. The work on dynamism and uncertainty can be extended to outdoor ambient assistive provision and smart urban city. This will induce more technical requirements that need to be handled which were omitted in this thesis, mainly the security aspects. Indeed, it is important to deal with user authentication and privacy during the build of context aware applications, especially in outdoor context where there is access to relevant data of the user profile.

Appendices

A GDS for assessment of primary degenerative dementia

Table A1: The global deterioration scale for assessment of primary degenerative dementia [13]

Level	Clinical Characteristics
1 No cognitive decline	No subjective complaints of memory deficit. No memory deficit evident on clinical interview.
2 Very mild cognitive decline (age-associated memory impairment)	Subjective complaints of memory deficit, most frequently in following areas: (a) forgetting where one has placed familiar objects; (b) forgetting names one formerly knew well. No objective evidence of memory deficit on clinical interview. No objective deficits in employment or social situations. Appropriate concern with respect to symptomatology.
3 Mild cognitive decline (Mild Cognitive Impairment)	Earliest clear-cut deficits. Manifestations in more than one of the following areas: (a) patient may have gotten lost when traveling to an unfamiliar location; (b) coworkers become aware of patient's relatively poor performance; (c) word and name finding deficit becomes evident to intimates; (d) patient may read a passage or a book and retain relatively little material; (e) patient may demonstrate decreased facility in remembering names upon introduction to new people; (f) patient may have lost or misplaced an object of value; (g) concentration deficit may be evident on clinical testing. Objective evidence of memory deficit obtained only with an intensive interview. Decreased performance in demanding employment and social settings. Denial begins to become manifest in patient. Mild to moderate anxiety accompanies symptoms.
4 Moderate cognitive decline (Mild Dementia)	Clear-cut deficit on careful clinical interview. Deficit manifest in following areas: (a) decreased knowledge of current and recent events; (b) may exhibit some deficit in memory of ones personal history; (c) concentration deficit elicited on serial subtractions; (d) decreased ability to travel, handle finances, etc. Frequently no deficit in following areas: (a) orientation to time and place; (b) recognition of familiar persons and faces; (c) ability to travel to familiar locations. Inability to perform complex tasks. Denial is dominant defense mechanism. Flattening of affect and withdrawal from challenging situations frequently occur.

5 Moderately severe cognitive decline (Moderate Dementia)	Patient can no longer survive without some assistance. Patient is unable during interview to recall a major relevant aspect of their current lives, e.g., an address or telephone number of many years, the names of close family members (such as grandchildren), the name of the high school or college from which they graduated. Frequently some disorientation to time (date, day of week, season, etc.) or to place. An educated person may have difficulty counting back from 40 by 4s or from 20 by 2s. Persons at this stage retain knowledge of many major facts regarding themselves and others. They invariably know their own names and generally know their spouses' and children's names. They require no assistance with toileting and eating, but may have some difficulty choosing the proper clothing to wear.
6 Severe cognitive decline (Moderately Severe Dementia)	May occasionally forget the name of the spouse upon whom they are entirely dependent for survival. Will be largely unaware of all recent events and experiences in their lives. Retain some knowledge of their past lives but this is very sketchy. Generally unaware of their surroundings, the year, the season, etc. May have difficulty counting from 10, both backward and, sometimes, forward. Will require some assistance with activities of daily living, e.g., may become incontinent, will require travel assistance but occasionally will be able to travel to familiar locations. Diurnal rhythm frequently disturbed. Almost always recall their own name. Frequently continue to be able to distinguish familiar from unfamiliar persons in their environment. Personality and emotional changes occur. These are quite variable and include: (a) delusional behavior, e.g., patients may accuse their spouse of being an impostor, may talk to imaginary figures in the environment, or to their own reflection in the mirror; (b) obsessive symptoms, e.g., person may continually repeat simple cleaning activities; (c) anxiety symptoms, agitation, and even previously nonexistent violent behavior may occur; (d) cognitive abulla, i.e., loss of willpower because an individual cannot carry a thought long enough to determine a purposeful course of action.
7 Very severe cognitive decline (Severe Dementia)	All verbal abilities are lost over the course of this stage. Frequently there is no speech at all -only unintelligible utterances and rare emergence of seemingly forgotten words and phrases. Incontinent of urine, requires assistance toileting and feeding. Basic psychomotor skills, e.g., ability to walk, are lost with the progression of this stage. The brain appears to no longer be able to tell the body what to do. Generalized rigidity and developmental neurologic reflexes are frequently present.

B Overview of Ambient Assisted Living

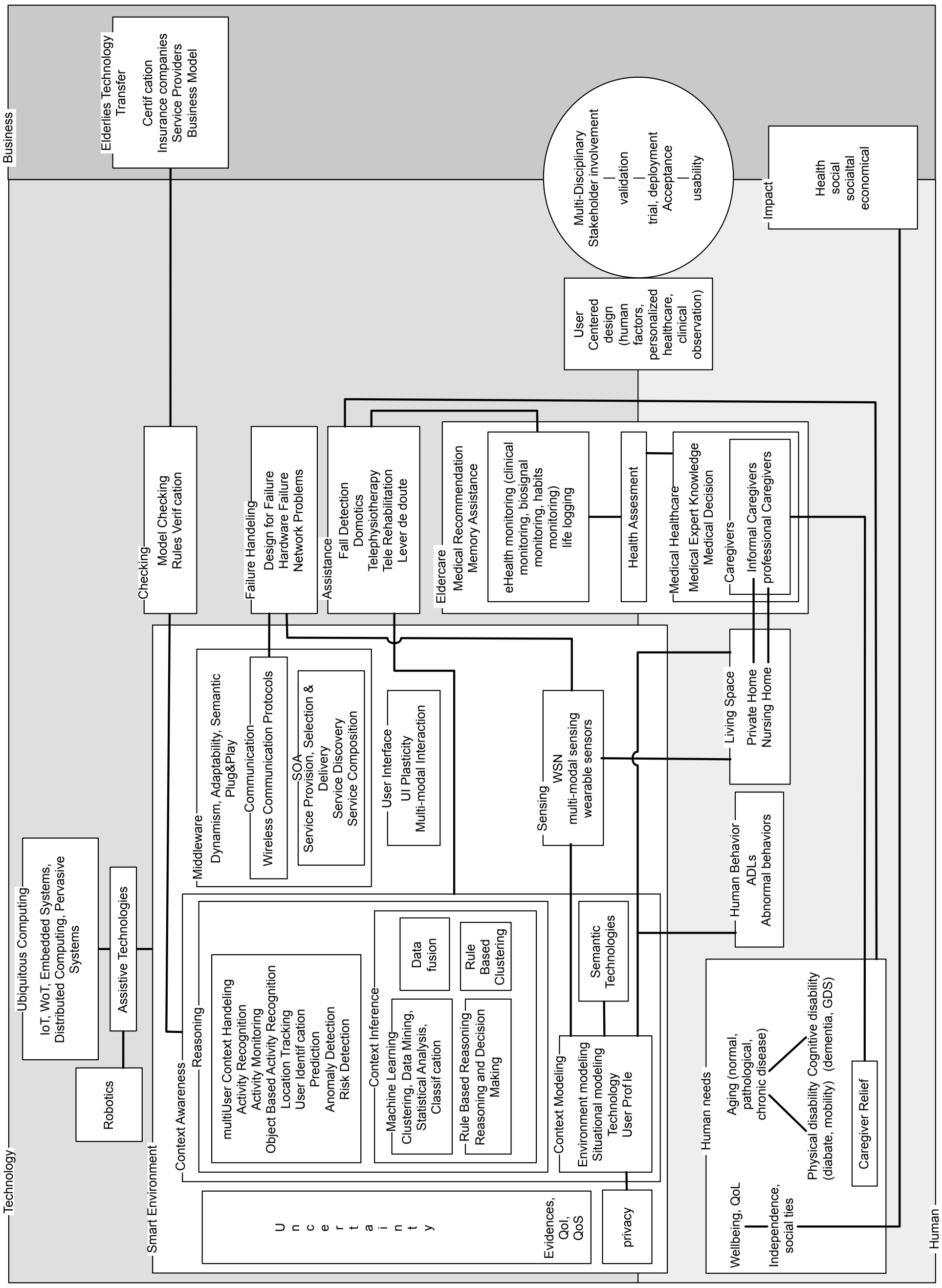


Figure B1: Overview of ambient assisted living

C The DPWS based semantic Plug&Play

In order to implement the mechanical Plug&Play mechanism described in Chapter 3, we have adapted the DPWS implementation used in the AMIGO[1] project, which is provided in open access. The *"Sensor Management System"* module is in charge of the auto-generation of the DPWS based bundle describing a new discovered sensor. Next is the Java method *"startSensor"* called once a new sensor is detected by the *"Sensor Management System"*:

```java
public static void startSensor(String name, String ID, String Events,
    double batterylevel) {
      //generate a new bundle with the new information received from
         Sensor Management System
      new BundleGeneration().generateBundle(name, ID, Events,
         batterylevel);
      //instantiate a service listener to detect the sensor bundle when
         it is started and communicate with it
      new SensorServiceListener().startListening(name + "-" + ID);
      //start the new sensor's bundle
      Bundle[] bundles = context.getBundles();
      for(Bundle b : bundles) {
           if(b.getSymbolicName() != null) {
                if(b.getSymbolicName().equals(name + "-" + ID)) {
                     try {
                          b.start();
                          System.out.println("[SMS DEBUG] Starting
                              sensor " + name + "-" + ID);
                     } catch (BundleException e) {
                          e.printStackTrace();
                     }
                }
           }
      }
}
```

[1]https://gforge.inria.fr/projects/amigo/

The auto-generation of new detected sensors' specific bundles consists in creating the *"config.properties"* and the *"MANIFEST.MF"* files specific for each of them, then packaging these files with the DPWS modules and the sensor's services Java classes. The *"config.properties"* and *"MANIFEST.MF"* files are represented below with the changing information highlighted in red:

In the *"config.properties"* file, we need to indicate the type and name of the sensor, its possible events, its battery level and the date of its installation.

```
## config.properties ##
#DPWSDevice
DPWSDevice.SCOPE=http://www.amupadh.com/AMUPADHProject
DPWSDevice.TYPE=SensorType
#DeviceModel
DPWSDevice.MANUFACTURERS=FranceTelecom
DPWSDevice.MANUFACTURER_URL=http://www.amupadh.com
DPWSDevice.MODEL_NAMES=PseudoDevice
DPWSDevice.MODEL_NUMBER=1.0
DPWSDevice.MODEL_URL=http://www.amupadh.com
DPWSDevice.PRESENTATION_URL=http://www.amupadh.com
#DEVICEMetadata
DPWSDevice.FIRMWARE_VERSION=1
DPWSDevice.FRIENDLY_NAMES=SensorID
DPWSDevice.SERIAL_NUMBER=1.0
DPWSDevice.EVENTS=SensorPossibleEvents
DPWSDevice.BATTERYLEVEL=SensorBatteryLevel
DPWSDevice.INSTALLDATE=SensorInstallationDate
#Class name of hosted Service in features
hostedServices=SensorUpdate
```

In the *"MANIFEST.MF"* file, we need to change the bundle name and symbolic name by the identification of the new sensor. This information will be used to identify the bundle in the OSGi framework.

```
Manifest-Version: 2.0
Bundle-ClassPath: ., lib/osgi.jar, lib/servlet-api-2.5.jar, lib/dpws-api
    -1.1-SNAPSHOT.jar, lib/osgi_R4_compendium-1.0.jar, lib/log4j-1.2.9.jar,
    lib/odonata-dpws-stack-1.0.6-SNAPSHOT.jar, lib/soda-jdom-1.0.jar
Bundle-Version: 1.0.0
Bundle-Name: SensorType-SensorID
Bundle-ManifestVersion: 2
Bundle-Activator: sg.ipal.pawm.aal.pseudoSensor.Activator
Import-Package: ...
Private-Package: ...
Bundle-SymbolicName: SensorType-SensorID
```

These changes are realized through the *"generateBundle"* method which generate the two files then combine them with other DPWS modules, and a common code for events forwarding for the different sensors. All these files are packaged to generate the sensor bundle. The generated bundle is installed on the sensors gateway OSGi container.

```
public void generateBundle(String Type, String ID, String Events,
    double batterylevel) {
  //Generate MANIFEST.MF file
  GenerateManifest(Type + "-" + ID);
  //Generate config.properties file
  GenerateConfig(Events, batterylevel);
  try {
    //Bundle jar generation
    Runtime rt = Runtime.getRuntime();
    Process pr = rt.exec("jar cfM WSMS/" + Type + "-" + ID + ".jar wsdl
        lib config.properties com META-INF");
    new BufferedReader(new InputStreamReader(pr.getInputStream()));
    pr.waitFor();
    System.out.println("bundle generated");
  } catch(Exception e) {
    System.out.println(e.toString());
    e.printStackTrace();
  }
  try {
```

```
    //Generated bundle installation
    Activator.context.installBundle("reference:file:WSMS/" + Type + "-"
        + ID + ".jar");
  } catch (BundleException e) {
    e.printStackTrace();
  }
}
```

The ge
discovery l
changes. S
and remov
bundle onc
of the gene

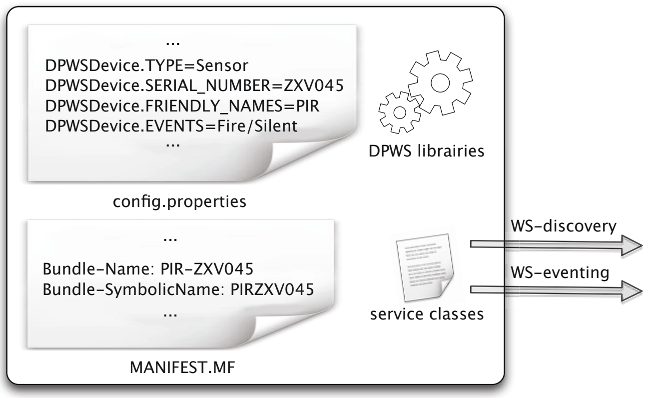


Figure C2: Structure of a PIR generated bundle

After the discovery of the new sensor by the OSGi assistive service container, the bundle representing the sensor on the OSGi sensors gateway starts sending the events received from the real sensor. This is achieved by the *"sendEvent"* method:

```
public void sendEvent(String state, String batterylevel) {
  //add the information that will be sent
  Namespace ns = Namespace.getNamespace("tns","http://www.amupadh.com/
      DPWS/SensorsGateway/Sensors");
```

```
  Element des = new Element("describtion", ns);
  Element elt = new Element("NewStatus", ns);
  elt.addContent(value.toUpperCase());
  des.addContent(elt);
  elt = new Element("BatteryLevel", ns);
  elt.addContent(batterylevel.toUpperCase());
  des.addContent(elt);
  //construct an XMLStreamReader to send the information
  JDOMStreamReader jdomsr = new JDOMStreamReader(des);
  XMLStreamReader[] xmlsr = new XMLStreamReader[1];
  xmlsr[0] = (XMLStreamReader) jdomsr;
  //send the new event
  serviceEventManager.postEvent("http://www.amupadh.com/DPWS/
     SensorsGateway/Sensors/UpdateSensorState/StatusChanged", xmlsr);
}
```

On the OSGi assistive service container, the *"Semantic Model Updater"* module detects new sensors' bundle through the discovery events, then use the method *"addSensor"* to invokes the registration information of the sensor and update the ontology.

```
public void addSensor() {
  //invoke the sensor's bundle to get its registration information
  MessageContent msgContent = pdc.invoke("http://www.amupadh.com/DPWS/
     SensorsGateway/Sensors/Register", null);
  //decode the invocation returned message
  if (msgContent != null) {
    try {
      FragmentStreamReader fsr = new FragmentStreamReader(msgContent.
          getBody()[0]);
      StaxBuilder builder = new StaxBuilder();
      org.jdom.Document doc = builder.build(fsr);
      Iterator<org.jdom.Element> iter;
      iter = (Iterator<org.jdom.Element>)doc.getRootElement().
          getChildren().iterator();
      while(iter.hasNext()) {
```

```
    org.jdom.Element elt = iter.next();
    if(elt.getName().equalsIgnoreCase("Status")) {
      status = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("Type")) {
      type = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("DiveceUUID")) {
      uuid = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("ID")) {
      id = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("BatteryLevel")) {
      batterylevel = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("Reliability")) {
      reliability = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("MaxLife")) {
      maxlife = elt.getText();
    } else if(elt.getName().equalsIgnoreCase("InstallDate")) {
      installdate = elt.getText();
    }
  }
  //create a new Sensor instance with the received information
  Sensor sensor = new Sensor(type + "-" + id, type, uuid, status,
      batterylevel, reliability, maxlife, installdate);
  //link the new instance to the sensor uuid
  sensors.put(uuid, sensor);
  //add sensor to the ontology
  n3Store.updateObject(HOME_NS + type, "a", MODEL_NS + "SensorType"
      );
  n3Store.updateObject(HOME_NS + type + id, "a", MODEL_NS + "Sensor
      ");
  n3Store.updateObject(HOME_NS + type + id, MODEL_NS + "hasType",
      HOME_NS + type);
  //add sensor possible states to the ontology
  for (String state : status.split("/")) {
    String triplestate = type + id + "_" + state;
    n3Store.updateObject(HOME_NS + triplestate, "a", MODEL_NS + "
        SensorState");
```

```
        n3Store.add(HOME_NS + type + id, MODEL_NS + "hasPossibleState",
            HOME_NS + triplestate);
      }
      //add sensor characteristics to the ontology
      n3Store.updateObject(HOME_NS + type + id, MODEL_NS + "
        hasBatteryLevel", batterylevel);
      n3Store.updateObject(HOME_NS + type + id, MODEL_NS + "
        hasMaxAutonomy", maxlife);
      n3Store.updateObject(HOME_NS + type + id, MODEL_NS + "
        hasReliability", reliability);
      ServiceReference ref = Activator.context.getServiceReference(
        DateR.class.getName());
      DateR dater = (DateR) Activator.context.getService(ref);
      n3Store.updateObject(HOME_NS + type + id, MODEL_NS + "
        hasInstallDate", dater.getN3Time(installdate));
    } catch (XMLStreamException e) {
      e.printStackTrace();
    }
  }
}
```

Sensor's properties are sent to the *"Configuration Tool"* in order to bind the new sensor to its environment and context. The Java method *"configureSensor"* is used to save the configuration from the graphic user interface to the ontology.

```
public void configureSensor(Sensor sensor, Map<String, String[]> guimap
    ) {
  //set the file where the semantic rules are written
  INFER_FILE = Activator.EULER_DIR + "infer-qol.n3";
  //get the sensor possible states
  List<String> states = sensor.getStates();
  //form the sensor name
  String sensorName = sensor.getName().split("-")[0] + sensor.getName()
      .split("-")[1];
  //add the semantic rule specific to each sensor possible event
```

```
  for (String state : states) {
    try {
      if(map.get(state)[0] != null && map.get(state)[1] != null && map.
          get(state)[2] != null) {
        BufferedWriter bw = new BufferedWriter(new FileWriter(
            INFER_FILE, true));
        bw.write(System.getProperty("line.separator"));
        //write the semantic rule generated from the gui to the
            inference file
        bw.write("{" + HOME_NS + sensorName + " qol:hasCurrentState [ a
             unc:Uncertainty; unc:uncertaintyLevel ?n; unc:
            relatedObject " + HOME_NS + sensorName + "_" + state.
            toUpperCase() + "]} => {ts:n3store ts:update {" + HOME_NS +
             map.get(state)[0] + " " + MODEL_NS + map.get(state)[1] + "
             [ a unc:Uncertainty; unc:uncertaintyLevel ?n; unc:
            relatedObject " + HOME_NS + map.get(state)[2] + "; unc:
            accordingTo " + HOME_NS + sensorName + "]}}.");
        bw.write(System.getProperty("line.separator"));
        bw.close() ;
      }
    } catch (IOException ex) {
      ex.printStackTrace();
    }
  }
}
```

Once the sensor use is configured, the *"Semantic Model Updater"* starts receiving the sensor's events and the reasoning is launched to infer new contexts. The Java method *"eventHandler"* in the *"Semantic Model Updater"* module receives sensor events and update the ontology.

```
public void eventHandler(Event event) {
  try {
    //transform the event to an XMLStreamReader
    XMLStreamReader[] body = (XMLStreamReader[]) ((MessageContent)event
        .getProperty(WSEventConstants.WS_EVENT_BODY)).getBody();
```

```
FragmentStreamReader fsr = new FragmentStreamReader(body[0]);
StaxBuilder builder = new StaxBuilder();
Document doc = builder.build(fsr);
Iterator<Element> iter = doc.getRootElement().getChildren().
    iterator();
//get the different elements of the event
while(iter.hasNext()) {
  Element elt = iter.next();
  if(elt.getName().equalsIgnoreCase("BatteryLevel")) {
    id = (String) event.getProperty("DPWS.device.id");
    batterylevel = elt.getText();
    Controller.getInstance().setBatteryLevel(id.split(":")[2],
        batterylevel);
  } else if(elt.getName().equalsIgnoreCase("NewStatus")) {
    state = elt.getText();
    id = (String) event.getProperty("DPWS.device.id");
    Controller.getInstance().setStatus(id.split(":")[2], state);
  }
}
Sensor se = Activator.control.sensors.get(id.split(":")[2]);
String sensorName = se.getName().split("-")[0] + se.getName().split
    ("-")[1];
//update the sensor batteryLevel
Activator.n3Store.updateObject(HOME_NS + sensorName , MODEL_NS + "
    hasBatteryLevel", batterylevel);
//get the ds mass function corresponding to the sensor
double ds_mass = Activator.massfunction.massFunctionClaculation(
    sensorName, state, time, batterylevel);
//update the sensor current state with uncertainty
Activator.n3Store.updateObject(HOME_NS + sensorName , MODEL_NS + "
    hasCurrentState", "[ a unc:Uncertainty; unc:uncertaintyLevel "
    + ds_mass + "; unc:relatedObject " +  HOME_NS + sensorName + "_
    " + state + "]");
//launch the reasoning
ServiceReference ref = Activator.context.getServiceReference(
    Reasoning.class.getName());
Reasoning rea = (Reasoning) Activator.context.getService(ref);
```

```
    rea.think();

  } catch (XMLStreamException e) {
    e.printStackTrace();
  }
}
```

D Semantic Modeling and Reasoning

Our reasoning is based on Euler[2] reasoning engine and the N3 syntax to create the ontology and write the semantic rules. Euler was packaged in the *"Euler Reasoner"* model and invoked by the *"Reasoning Engine"* module when the context changes. Euler uses different files passed as arguments when it is executed. We have defined three principle files which are used by Euler: load-model.n3, load-uncertainty.n3 and infer.n3.

The load-model.n3 file presented below contains all the classes and properties of the ontology.

```
## CLASSES ##
aal:Person a rdfs:Class.
aal:Resident a rdfs:Class; rdfs:subClassOf aal:Person.
aal:Caregiver a rdfs:Class; rdfs:subClassOf aal:Person.
aal:Environment a rdfs:Class.
aal:House a rdfs:Class; rdfs:subClassOf aal:Environment.
aal:Outside a rdfs:Class; rdfs:subClassOf aal:Environment.
aal:Room a rdfs:Class; rdfs:subClassOf aal:Environment.
aal:Bedroom a rdfs:Class; rdfs:subClassOf aal:Room.
aal:Livingroom a rdfs:Class; rdfs:subClassOf aal:Room.
aal:Kitchen a rdfs:Class; rdfs:subClassOf aal:Room.
aal:Bathroom a rdfs:Class; rdfs:subClassOf aal:Room.
aal:Toilet a rdfs:Class; rdfs:subClassOf aal:Room.
aal:Object a rdfs:Class.
aal:Furniture a rdfs:Class; rdfs:subClassOf aal:Object.
aal:Sensor a rdfs:Class.
aal:SensorState a rdfs:Class.
aal:SensorType a rdfs:Class.
aal:Device a rdfs:Class.
aal:Service a rdfs:Class.
aal:Activity a rdfs:Class.
aal:Deviance a rdfs:Class; rdfs:subClassOf aal:Activity;
  rdfs:comment "Problematic activity"@en.
aal:Normal a rdfs:Class; rdfs:subClassOf aal:Activity.
## OBJECT PROPERTIES ##
```

[2]http://eulersharp.sourceforge.net

```
aal:liveIn a rdf:ObjectProperty;
  rdfs:comment "House where the resident live."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Environment.
aal:detectedIn a rdf:ObjectProperty;
  rdfs:comment "Room where the resident is detected."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Environment.
aal:useNow a rdf:ObjectProperty;
  rdfs:comment "Object a person is currently using."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Object.
aal:believedToDo a rdf:ObjectProperty;
  rdfs:comment "Activity a resident is believed to be doing."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Activity.
aal:cameFrom a rdf:ObjectProperty;
  rdfs:comment "Room the resident was in before the current one."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Environment.
aal:partOf a rdf:ObjectProperty;
  a owl:TransitiveProperty;
  rdfs:comment "Describe inclusion of environments."@en;
  rdfs:domain aal:Environment; rdfs:range aal:Environment.
aal:locatedIn a rdf:ObjectProperty;
  rdfs:comment "Location of a door in the environment."@en;
  rdfs:domain aal:Object; rdfs:range aal:Environment.
aal:deployedIn a rdf:ObjectProperty;
  rdfs:comment "Deployment location of a sensor."@en;
  rdfs:domain aal:Sensor; rdfs:range aal:Environment.
aal:attachedTo a rdf:ObjectProperty;
  rdfs:comment "Describe the binding of sensor to a furniture."@en;
  rdfs:domain aal:Sensor; rdfs:range aal:Object.
aal:hasPossibleState a rdf:ObjectProperty;
  rdfs:comment "Possible state of a sensor."@en;
  rdfs:domain aal:Sensor; rdfs:range aal:SensorState.
aal:hasType a rdf:ObjectProperty;
  rdfs:comment "Type of a sensor."@en;
  rdfs:domain aal:Sensor; rdfs:range aal:SensorType.
aal:hasCurrentState a rdf:ObjectProperty;
  rdfs:comment "Current state of a sensor."@en;
  rdfs:domain aal:Sensor; rdfs:range aal:SensorState.
```

```
aal:needService a rdf:ObjectProperty;
  rdfs:comment "A deviance activity need assistive service."@en;
  rdfs:domain aal:Deviance; rdfs:range aal:Service.
aal:isCorrectionOf a rdf:ObjectProperty;
  rdfs:comment "A normal activity is a correction of a deviant activity
      ."@en;
  rdfs:domain aal:Normal; rdfs:range aal:Deviance.
aal:interestedIn a rdf:ObjectProperty;
  rdfs:comment "the patient needs an assistive service."@en;
  rdfs:domain aal:Resident; rdfs:range aal:Service.
aal:inChargeOf a rdf:ObjectProperty;
  rdfs:comment "A caregiver is incharge of the resident."@en;
  rdfs:domain aal:Caregiver; rdfs:range aal:Resident.
aal:useDevice a rdf:ObjectProperty;
  rdfs:comment "A person is using a device."@en;
  rdfs:domain aal:Person; rdfs:range aal:Device.
## DATATYPE PROPERTIES ##
aal:hasId a rdf:DatatypeProperty;
  rdfs:comment "Sensor identification."@en;
  rdfs:domain aal:Sensor; rdfs:range xsd:string.
aal:hasReliability a rdf:DatatypeProperty;
  rdfs:comment "Sensor reliability."@en;
  rdfs:domain aal:Sensor; rdfs:range xsd:double.
aal:hasMaxAutonomy a rdf:DatatypeProperty;
  rdfs:comment "Sensor reliability."@en;
  rdfs:domain aal:Sensor; rdfs:range xsd:int.
aal:hasInstallDate a rdf:DatatypeProperty;
  rdfs:comment "Indicate the id of the sensor."@en;
  rdfs:domain aal:Sensor; rdfs:range xsd:date.
aal:hasBatteryLevel a rdf:DatatypeProperty;
  rdfs:comment "Sensor reliability."@en;
  rdfs:domain aal:Sensor; rdfs:range xsd:double.
aal:isAlone a rdf:DatatypeProperty;
  rdfs:comment "Is the resident alone in the environment?"@en;
  rdfs:domain aal:Resident; rdfs:range xsd:boolean.
aal:inRoomSince a rdf:DatatypeProperty;
  rdfs:comment "The time when the resident entered his current location
```

```
    "@en;
  rdfs:domain aal:Resident; rdfs:range xsd:dateTime.
aal:inRoomFor a rdf:DatatypeProperty;
  rdfs:comment "The duration since the resident entered his current
    location, in seconds"@en;
  rdfs:domain aal:Resident; rdfs:range xsd:duration.
aal:doesActivitySince a rdf:DatatypeProperty;
  rdfs:comment "The time when the resident supposedly started an
    activity"@en;
  rdfs:domain aal:Resident; rdfs:range xsd:dateTime.
aal:doesActivityFor a rdf:DatatypeProperty;
  rdfs:comment "The duration since the resident supposedly started an
    activity, in seconds"@en;
  rdfs:domain aal:Resident; rdfs:range xsd:duration.
aal:motionMeasured a rdf:DatatypeProperty;
  rdfs:comment "Measurement of the number of sensor activations in a
    given space during a given time window."@en;
  rdfs:domain aal:Environment;  rdfs:range xsd:int.
aal:hasValue a rdf:DatatypeProperty;
  rdfs:comment "value provided by the sensors which dont have fixed
    state."@en;
  rdfs:domain aal:Sensor.
aal:lastUpdate a rdf:DatatypeProperty;
  rdfs:comment "Date and time of the last update of a sensor state."@en
    ;
  rdfs:domain aal:Sensor; rdfs:range xsd:dateTime.
aal:lastUsed a rdf:DatatypeProperty;
  rdfs:comment "Date and time of the last time an object was used."@en;
  rdfs:domain aal:Object; rdfs:range xsd:dateTime.
aal:notUsedFor a rdf:DatatypeProperty;
  rdfs:comment "Duration since an object was last used."@en;
  rdfs:domain aal:Object; rdfs:range xsd:duration.
aal:hasLastUpdate a rdf:DatatypeProperty;
  rdfs:comment "Indicate whether the sensor is the last one updated."
    @en;
  rdfs:domain aal:Sensor; rdfs:range xsd:boolean.
aal:indicateLocation a rdf:DatatypeProperty;
```

```
  rdfs:comment "Whether SensorState indicate the resident location."@en
      ;
  rdfs:domain aal:SensorState;  rdfs:range xsd:boolean.
aal:indicateUse a rdf:DatatypeProperty;
  rdfs:comment "Whether SensorState indicate the use of an object."@en;
  rdfs:domain aal:SensorState;  rdfs:range xsd:boolean.
aal:hasMaxWanderingDuration a rdf:DatatypeProperty;
  rdfs:comment "Maximum duration to wait before starting the wandering
      alert service."@en;
  rdfs:domain aal:Resident; rdfs:range xsd:int.
aal:hasShowerDuration a rdf:DatatypeProperty;
  rdfs:comment "Maximum duration of a resident shower."@en;
  rdfs:domain aal:Resident; rdfs:range xsd:int.
```

The load-model.n3 file loads the uncertainty model used to integrate uncertain information. It comports mainly the Uncertainty class and the different properties relatedObject, accordingTo and uncertaintyLevel:

```
@prefix rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rdfs:   <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl:    <http://www.w3.org/2002/07/owl#>.
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#>.
@prefix unc:  <load-uncertainty#>.
## CLASSES ##
unc:Uncertainty a rdfs:Class.
## OBJECT PROPERTIES ##
unc:relatedObject a rdf:ObjectProperty;
  rdfs:comment "Define the range of the original property without
      uncertainty."@en;
  rdfs:domain unc:Uncertainty.
unc:accordingTo a rdf:ObjectProperty;
  rdfs:comment "Define the range of the original property without
      uncertainty."@en;
  rdfs:domain unc:Uncertainty.
## DATATYPE PROPERTIES ##
```

```
unc:uncertaintyLevel a owl:DatatypeProperty;
  rdfs:comment "define the degree of uncertainty of the property."@en;
  rdfs:domain unc:Uncertainty;
  rdfs:range xsd:double.
```

The different rules used in our reasoning are included in the infer.n3 file, mainly the rules for assistive service and interaction device selection and the rules for abnormal activities detection in order to fire the rule to start the assistive service.

```
## RULES ##

## Infer services to start ##
{?r aal:believedToDo ?d. ?d aal:needService ?s. <input.n3>!log:
    semantics log:notIncludes {?r aal:interestedIn ?s}} => {ts:n3store
    ts:update {?r aal:interestedIn ?s}}.

## Infer services to stop ##
{?r aal:interestedIn ?s. ?d aal:needService ?s. ?a aal:isCorrectionOf ?
    d. ?r aal:believedToDo ?a} => {ts:n3store ts:remove {?r aal:
    interestedIn ?s}}.

## Infer Device to use ##
{?r aal:detectedIn ?l. ?d ske:deployedIn ?l} => {ts:n3store ts:update
    {:r ske:useDevice ?d}}.

## tracks resident location in the house ##
{?se aal:hasCurrentState ?st. ?se aal:hasLastUpdate true. ?st aal:
    indicateLocation true. ?se aal:deployedIn ?r. ?r aal:partOf ?h. ?u
    aal:liveIn ?h. ?u aal:detectedIn ?r2. ?r log:notEqualTo ?r2. ?se
    aal:lastUpdate ?t} => {ts:n3store ts:update {?u aal:detectedIn ?r.
    ?u aal:cameFrom ?r2. ?u aal:inRoomSince ?t}}.

## update uncertainty of resident location after each event ##
{?se aal:hasCurrentState [ a unc:Uncertainty; unc:uncertaintyLevel ?n;
    unc:relatedObject ?st]. ?se aal:hasLastUpdate true. ?st aal:
```

```
    indicateLocation true. ?se aal:deployedIn ?r. ?r aal:partOf ?h. ?u
    aal:liveIn ?h} => {ts:n3store ts:update {?u aal:detectedIn [ a unc:
    Uncertainty; unc:uncertaintyLevel ?n; unc:relatedObject ?r; unc:
    accordingTo ?se]}}.

## tracks resident move in the house with uncertainty ##
{?se aal:hasCurrentState [ a unc:Uncertainty; unc:uncertaintyLevel ?n;
    unc:relatedObject ?st]. ?se aal:hasLastUpdate true. ?st aal:
    indicateLocation true. ?se aal:deployedIn ?r. ?r aal:partOf ?h. ?u
    aal:liveIn ?h. ?u aal:detectedIn [ a unc:Uncertainty; unc:
    uncertaintyLevel ?n2; unc:relatedObject ?r2]. ?r log:notEqualTo ?r2
    . ?se aal:lastUpdate ?t} => {ts:n3store ts:update {?u aal:cameFrom
    ?r2. ?u aal:inRoomSince ?t}}.

## infer duration of activities ##
{?x aal:doesActivitySince ?start. hom:clock aal:hasValue ?now. (?now ?
    start) math:difference ?duration} => {?x aal:doesActivityFor ?
    duration}.

## Wandering at night service ##
{?se aal:hasType aal:pressure. ?se aal:hasLastUpdate true. ?se aal:
    deployedIn aal:Bedroom. aal:Bedroom aal:partOf ?h. ?u aal:liveIn ?h
    . ?se aal:hasCurrentState {?se string:concatenation "-silent"}. hom
    :clock aal:hasValue ?now} => {ts:n3store ts:update {?u aal:
    believedToDo aal:notInTheBed. ?u aal:doesActivitySince ?now}}.

{?u aal:detectedIn ?r. ?u aal:believedToDo aal:notInTheBed ?u. aal:
    doesActivityFor ?d. ?u aal:hasMaxWanderingDuration ?duration. ?d
    math:notLessThan ?duration} => {ts:n3store ts:update {?u aal:
    believedToDo aal:wanderingAtNight}}.

## Wandering at night service with uncertainty ##
{?se aal:hasType aal:pressure. ?se aal:hasLastUpdate true. ?se aal:
    deployedIn aal:Bedroom. aal:Bedroom aal:partOf ?h. ?u aal:liveIn ?h
    . ?se aal:hasCurrentState [ a unc:Uncertainty; unc:uncertaintyLevel
     ?n; unc:relatedObject {?se string:concatenation "-silent"}]. hom:
    clock aal:hasValue ?now} => {ts:n3store ts:update {?u aal:
```

```
   believedToDo [ a unc:Uncertainty; unc:uncertaintyLevel ?n; unc:
   relatedObject aal:notInTheBed; unc:accordingTo ?se]. ?u aal:
   doesActivitySince ?now}}.

{?u aal:detectedIn [ a unc:Uncertainty; unc:uncertaintyLevel ?n1; unc:
   relatedObject ?r; unc:accordingTo ?se1]. ?u aal:believedToDo [ a
   unc:Uncertainty; unc:uncertaintyLevel ?n2; unc:relatedObject aal:
   notInTheBed; unc:accordingTo ?se2]. aal:doesActivityFor ?d. ?u aal:
   hasMaxWanderingDuration ?duration. ?d math:notLessThan ?duration}
   => {ts:n3store ts:update {?u aal:believedToDo [ a unc:Uncertainty;
   unc:uncertaintyLevel {{?n1 math:sum ?n2} math:quotient 2}; unc:
   relatedObject aal:wanderingAtNight; unc:accordingTo ?se1; unc:
   accordingTo ?se2]}}.

## Shower for too long service ##
{?se aal:hasType aal:vibrator. ?se aal:hasLastUpdate true. ?se aal:
   deployedIn aal:Bathroom. aal:Bathroom aal:partOf ?h. ?u aal:liveIn
   ?h. ?se aal:hasCurrentState {?se string:concatenation "-
   unstationary"}. hom:clock aal:hasValue ?now} => {ts:n3store ts:
   update {?u aal:believedToDo aal:TakingShower. ?u aal:
   doesActivitySince ?now}}.

{?u aal:detectedIn ?r. ?u aal:believedToDo aal:TakingShower. ?u aal:
   doesActivityFor ?d. ?u aal:hasShowerDuration ?duration. ?d math:
   notLessThan ?duration} => {ts:n3store ts:update {?u aal:
   believedToDo aal:ShoweringTooLong}}.

## Shower for too long service with uncertainty ##
{?se aal:hasType aal:vibrator. ?se aal:hasLastUpdate true. ?se aal:
   deployedIn aal:Bathroom. aal:Bathroom aal:partOf ?h. ?u aal:liveIn
   ?h. ?se aal:hasCurrentState [ a unc:Uncertainty; unc:
   uncertaintyLevel ?n; unc:relatedObject {?se string:concatenation "-
   unstationary"}]. hom:clock aal:hasValue ?now} => {ts:n3store ts:
   update {?u aal:believedToDo [ a unc:Uncertainty; unc:
   uncertaintyLevel ?n; unc:relatedObject aal:TakingShower; unc:
   accordingTo ?se]. ?u aal:doesActivitySince ?now}}.
```

```
{?u aal:detectedIn [ a unc:Uncertainty; unc:uncertaintyLevel ?n1; unc:
   relatedObject ?r; unc:accordingTo ?se1]. ?u aal:believedToDo [ a
   unc:Uncertainty; unc:uncertaintyLevel ?n1; unc:relatedObject aal:
   TakingShower; unc:accordingTo ?se2]. ?u aal:doesActivityFor ?d. ?u
   aal:hasShowerDuration ?duration. ?d math:notLessThan ?duration} =>
   {ts:n3store ts:update {?u aal:believedToDo [ a unc:Uncertainty; unc
   :uncertaintyLevel {{?n1 math:sum ?n2} math:quotient 2}; unc:
   relatedObject aal:ShoweringTooLong; unc:accordingTo ?se1; unc:
   accordingTo ?se2]}}.

## Tap left on service ##
{?se aal:hasType aal:vibrator. ?se aal:hasLastUpdate true. ?se aal:
   attachedTo aal:pipeline. ?u aal:detectedIn ?r. ?r aal:partOf ?h. ?u
    aal:liveIn ?h. ?se aal:hasCurrentState {?se string:concatenation "
   -unstationary"}. hom:clock aal:hasValue ?now} => {ts:n3store ts:
   update {?u aal:believedToDo aal:UsingTap. ?u aal:doesActivitySince
   ?now}}.

{?u aal:detectedIn ?r. ?r log:notEqualTo aal:Bathroom. ?u aal:
   believedToDo aal:UsingTap. ?u aal:doesActivityFor ?d. ?d math:
   notLessThan 300} => {ts:n3store ts:update {?u aal:believedToDo aal:
   TapLeftOn}}.

## Tap left on service with uncertainty ##
{?se1 aal:hasType aal:vibrator. ?se aal:hasLastUpdate true. ?se aal:
   attachedTo aal:pipeline. ?r aal:partOf ?h. ?u aal:liveIn ?h. ?se
   aal:hasCurrentState [ a unc:Uncertainty; unc:uncertaintyLevel ?n1;
   unc:relatedObject {?se string:concatenation "-unstationary"}]. ?u
   aal:detectedIn [ a unc:Uncertainty; unc:uncertaintyLevel ?n2; unc:
   relatedObject ?r; unc:accordingTo ?se2]. hom:clock aal:hasValue ?
   now} => {ts:n3store ts:update {?u aal:believedToDo [ a unc:
   Uncertainty; unc:uncertaintyLevel {{?n1 math:sum ?n2} math:quotient
    2}; unc:relatedObject  aal:UsingTap; unc:accordingTo ?se1; unc:
   accordingTo ?se2]. ?u aal:doesActivitySince ?now}}.

{?u aal:detectedIn [ a unc:Uncertainty; unc:uncertaintyLevel ?n1; unc:
   relatedObject ?r; unc:accordingTo ?se1]. ?r log:notEqualTo aal:
```

```
Bathroom. ?u aal:believedToDo [ a unc:Uncertainty; unc:
uncertaintyLevel n2; unc:relatedObject  aal:UsingTap; unc:
accordingTo ?se1; unc:accordingTo ?se2]. ?u aal:doesActivityFor ?d.
 ?d math:notLessThan 300} => {ts:n3store ts:update {?u aal:
believedToDo [ a unc:Uncertainty; unc:uncertaintyLevel {{?n1 math:
sum ?n2} math:quotient 2}; unc:relatedObject  aal:TapLeftOn; unc:
accordingTo ?se1; unc:accordingTo ?se2]}}.
```

Other files have also been used such as:

- load-home.n3 file which instantiates the model with the description of each room and its residents
- load-init.n3 file which contains some initiation information.
- infer-logger.n3 file containing some rules that help to log interesting reasoning output.
- infer-triplestore.n3 file which contains the object-properties used to manage the nTripleStore we have created. The nTripleStore is a kind of dataset where all the triples of the ontology are stored. The use of the triple store helped to reduce the reasoning time as we load the files only when the framework is started instead of reloading it each time to reason. Modifications are integrated in the triple store.
- dump.n3 file is generated after each reasoning with the updated ontology.

E Uncertainty Handling

Two main parts have been added to the UbiSMART framework in order to handle the uncertainty about sensor events derived from their technical and functional characteristics. These two parts are the measurement of uncertainty and the decision making in conflicting situations. Other parts have been updated in order to take into consideration the uncertainty about event such as the *"Configuration Tool"*, the *"Semantic Model Updater"* and the *"Thought Interpreter"* modules.

For the measurement of uncertainty, two modules have been created, the *"Uncertainty Measurement"* module and the *Functional Param"* module. In the *"Uncertainty Measurement"* module, the *"massFunctionCalculation"* method is used to calculate the uncertainty about the current sensor event:

```
public double massFunctionCalculation(String sensor, String value,
    String time, String batterylevel) {
  // a vector for all the characteristic parameters of the current
     sensor
  Vector<Double> CharacteristicParameters = new Vector<Double>();
  // add the battery level percentage characteristic parameter
  CharacteristicParameters.add(Double.parseDouble(batterylevel));
  // update the fiability of the sensor
  FiabilityCalculator fiabCalculator = new FiabilityCalculator();
  double fiability = fiabCalculator.updateFiability(sensor, value, time
     );
  // add the fiability percentage characteristic parameter
  CharacteristicParameters.add(fiability);
  // update the current autonomy of the sensor
  AutonomyCalculator autoCalculator = new AutonomyCalculator();
  double autonomy = autoCalculator.updateAutonomy(sensor, value, time);
  // add the autonomy percentage characteristic parameter
  CharacteristicParameters.add(autonomy);
  // calculate the functional parameter
  ServiceReference ref = Activator.bc.getServiceReference(
     FunctionalParamCalculator.class.getName());
  FunctionalParamCalcuator funcparam = (FunctionalParamCalculator)
     Activator.bc.getService(ref);
```

```
  double functionalparam = funcparam.functionalParamCalculation(sensor,
      value);
  // calculate the uncertainty about the sensor event
  double ds_mass = 1;
  for (Double cParameter : CharacteristicParameters) {
    ds_mass = ds_mass * ((double)cParameter /(double)(37 * (Math.expm1(
        cParameter/100) + 1)));
  }
  ds_mass = ds_mass * (functionalparam/100);

  return ds_mass;
}
```

The *"massFunctionCalculation"* Java method uses two other methods to update the reliability and estimated remaining life time of the sensor. It also invokes the remote method *"functionalParamCalculation"* from the *"Functional Param"* module to get the value of the functional parameter I have identified for our use-case.

```
public double functionalParamCalculation( String sensor, String value)
    {

  double functionalparam = 0;
  if(value.equalsIgnoreCase("on"))
    //increment the number of successive event for the current sensor
    incrementSuccessiveEvents(sensor);
  else {
    //calculate the functional param as defined by the equation
    if(totalSuccessiveEventList.get(sensor)<maxSuccessiveEventList.get(
        sensor)+(maxSuccessiveEventList.get(sensor)/2))
          functionalparam = 100 - Math.pow(((totalSuccessiveEventList.
              get(sensor)*2.16)/(maxSuccessiveEventList.get(sensor)+(
              maxSuccessiveEventList.get(sensor)/2))),6);
         else
          functionalparam = 0.0;
    //reinitialize to zero the number of successive events for the
        current sensor
```

```
    reinitializeSuccessiveEvents(sensor);
  }
        return functionalparam;
}
```

On the decision making part, I have integrated a library for Dempster-Shafer calculation into the *"DST Decision Making"* module [3]. This module is composed of a principal method *"decisionMaking"* and different supporting methods.

```
public String decisionMaking(String result) {
  //decode the result received from the thought interpreter
  String subject = result.split("\\{")[1].split(" ")[0].split(":")[1];
  String predicate = result.split("\\{")[1].split(" ")[1].split(":")
     [1];
  String object = result.split("unc:relatedObject ")[1].split("\\.")
     [0].split(":")[1];
  String accordingto = result.split("unc:accordingTo ")[1].split("}")
     [0].split(":")[1];
  String uncertaintylevel = result.split("unc:uncertaintyLevel ")[1].
     split(" \\.")[0];
  //add a new mass function corresponding to the sensor and the
     predicate
  String mfkey = "m_" + accordingto + "_" + predicate;
  if(!DSTCalculation.massFunctionList.containsKey(mfkey))
    DSTCalculation.massFunctionList.put(mfkey, new MassFunction<String
       >());

  DSTCalculation dstcalcul = new DSTCalculation();
  //get all the possible hypotheses related to the predicate
  dstcalcul.getHypotheses(predicate);
  //add values to the different mass functions
  dstcalcul.createMassFunction(object, mfkey, predicate,
     uncertaintylevel);
  //calculate a joint mass function for the different sensors
```

[3] http://sourceforge.net/projects/jds/?source=navbar

```
  MassFunction<String> jointDistribution = dstcalcul.
     fusionMassFunctionCalculation(predicate);
  //choose the most plausible hypothesis
  String mostplau = jointDistribution.getMostPlausibleSingletons().
     toString();
  String objectdecision = mostplau.substring(1,mostplau.length()-1);
  // return a triple as a decision
  String decision = subject + " " + predicate + " " + objectdecision;
        return decision;
}
```

The *"getHypotheses"* Java method fetches the ontology to get all the possible hypotheses related to the conflicting predicate.

```
public void getHypotheses(String predicate){

  //use the triple store to get the rang of the predicate
  ServiceReference ref = Activator.bc.getServiceReference(NTripleStore.
     class.getName());
  NTripleStore n3Store = (NTripleStore) Activator.bc.getService(ref);
  String rang = (String)n3Store.searchURIs("qol:" + predicate, "rdfs:
     range" , "?").toArray()[0];
  //use the triple store to get all the instances of the rang of the
     predicate
  Collection<String> instances = n3Store.searchURIs("?", "a", rang);
  for (String instance: instances) {
    //create a hypothesis for each instance
    String hypothesis = "h_" + instance.split(":")[1];
    Hypothesis<String> hypothesisObj = new Hypothesis<String>(instance.
       split(":")[1]);
    //add the new hypothesis to masshypothesesList
    if(masshypothesesList.containsKey(predicate))
      masshypothesesList.get(predicate).put(hypothesis, hypothesisObj);
    else {
      HashMap<String, Hypothesis<String>> temp = new HashMap<String,
         Hypothesis<String>>();
```

```
      temp.put(hypothesis, hypothesisObj);
      masshypothesesList.put(predicate, temp);
    }
    //add the new hypothesis to hypothesesList
    if(!hypothesesList.containsKey(hypothesis))
      hypothesesList.put(hypothesis, hypothesisObj);
  }
}
```

The *"createMassFunction"* Java method creates and assigns values to the mass functions of the different possible hypothesis.

```
public void createMassFunction(String object, String mfkey, String
    predicate, String uncertaintylevel) {
  //get the hypothesis and the mass function relating the object and
      predicate
  Hypothesis<String> hypothese = hypothesesList.get("h_" + object);
  MassFunction<String> massfunction = massFunctionList.get(mfkey);
  //assign uncertaintylevel to the mass function and its related
      hypothesis
  massfunction.remove(hypothese);
  massfunction.add(hypothese, Double.parseDouble(uncertaintylevel));
  //assign 0 to all the other hypotheses
  HashMap<String, Hypothesis<String>> hypothesesList =
      masshypothesesList.get(predicate);
  for(String hypothesiskey : hypothesesList.keySet()) {
    if(!massfunction.getFrameOfDiscernment().contains(hypothesiskey))
      massfunction.add(hypothesesList.get(hypothesiskey), 0);
  }
  //assign the rest to the ignorance set
  if(massfunction.getFrameOfDiscernment().size() > 1) {
    double t = 1 - massfunction.getMassSum();
    massfunction.add(massfunction.getFrameOfDiscernment(), t);
  }
}
```

Finally, the *"fusionMassFunctionCalculation"* Java method uses the combination rule of DST to calculate the combination of the different mass functions created.

```
public MassFunction<String> fusionMassFunctionCalculation(String
    predicate) {
  Set<String> keysSet = massFunctionList.keySet();
  Vector<MassFunction<String>> fusionMassFunctionList = new Vector<
      MassFunction<String>>();
  //put all the mass functions related to the predicate into the
      fusionMassFunctionList
  for (String key : keysSet) {
    if(key.contains(predicate) && massFunctionList.get(key).getMassSum
        () > 0)
      fusionMassFunctionList.add(massFunctionList.get(key));
  }
  MassFunction<String> jointDistribution = new MassFunction<String>();
  //apply the combination rule of DST if the number of mass functions
      is higher than 1
  if(fusionMassFunctionList.size()>1) {
    jointDistribution = fusionMassFunctionList.elementAt(0).
        combineConjunctive(fusionMassFunctionList.elementAt(1));
    for (int k = 2; k < fusionMassFunctionList.size(); k++)
      jointDistribution = jointDistribution.combineConjunctive(
          fusionMassFunctionList.elementAt(k));
    System.out.println("Dempster joint distrubtion for all sensors:" +
        jointDistribution);
  }
  return jointDistribution;
}
```

List of Figures

List of Tables

Bibliography

[1] D. Steven and B. Patrice, "Doing more with less: The aging population challenge," *ARM Medical Special*.

[2] J. Birren and V. Renner, "Research on the psychology of ageing," *Handbook of psychology of ageing, New York: Van Nostrand Reinhold*, 1977.

[3] F. I. F. O. A. R. Statistics, "Older americans 2010." http://www.agingstats.gov/agingstatsdotnet/Main_Site/Data/2010_Documents/Docs/OA_2010.pdf, 2010.

[4] S. Jacobzone, "Ageing and care for frail elderly persons: an overview of international perspectives," *OECD Labour Market and Social Policy Occasional Papers*, 1999.

[5] W. Mann, "The aging population and its needs," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 12–14, 2004.

[6] D. o. E. United Nations Secretariat and P. D. Social Affairs, "World population prospects: The 2010 revision." http://esa.un.org/unpd/wpp/index.htm, 2010.

[7] W. E. Forum, "Global population ageing: Peril or promise?." http://www3.weforum.org/docs/WEF_GAC_GlobalPopulationAgeing_Report_2012.pdf, 2012.

[8] B. Bonder and M. Wagner, *Functional performance in older adults.* FA Davis Company, 2009.

[9] J. Bischkopf, A. Busse, and M. Angermeyer, "Mild cognitive impairment1–a review of prevalence, incidence and outcome according to current approaches," *Acta Psychiatrica Scandinavica*, vol. 106, no. 6, pp. 403–414, 2002.

[10] A. Wimo and M. Prince, *World Alzheimer Report 2010: the global economic impact of dementia.* Alzheimer's Disease International, 2010.

[11] M. Prince and J. Jackson, *World Alzheimer Report 2009.* Alzheimer's Disease International, 2009.

[12] J. Diamond, *A report on Alzheimer's disease and current research.* Alzheimer Society of Canada, 2006.

[13] B. Reisberg, S. Ferris, M. de Leon, and T. Crook, "The global deterioration scale for assessment of primary degenerative dementia.," *The American journal of psychiatry*, 1982.

[14] H. Pigot, A. Mayers, and S. Giroux, "The intelligent habitat and everyday life activity support," in *5th international conference on Simulations in Biomedicine*, pp. 507–516, 2003.

[15] K. Fletcher, "Nursing standard of practice protocol: Recognition and management of dementia," *Hartford Institute for Geriatric Nursing.[online]. January*, 2008.

[16] R. Orpwood, C. Gibbs, T. Adlam, R. Faulkner, and D. Meegahawatte, "The design of smart homes for people with dementia-user-interface aspects," *Universal Access in the Information Society*, vol. 4, no. 2, pp. 156–164, 2005.

[17] I. Mebane-Sims, "2009 alzheimer's disease facts and figures.," *Alzheimer's & Dementia*, 2009.

[18] T. Harrington and M. K. Harrington, *Gerontechnology: Why and how.* Shaker Maastricht, 2000.

[19] M. Weiser, "The computer for the twenty-first century (pp. 94-100)," *Scientific American, September Issue*, 1991.

[20] H. Nakashima, H. Aghajan, and J. Augusto, *Handbook of ambient intelligence and smart environments.* Springer, 2009.

[21] E. Aarts and S. Marzano, "The new everyday: Visions of ambient intelligence, 010 publishing," *Rotterdam, The Netherlands*, vol. 116, 2003.

[22] H. Aldred, T. Amaral, S. Brownsell, J. L. Arnott, M. Hawley, and N. Hine, "Supporting older people through telecare," in *Proc. 8th Conf. on Advancement of Assistive Technology in Europe (AAATE 05)*, pp. 41–45, 2005.

[23] J. Hanson, D. Osipovič, N. Hine, T. Amaral, R. Curry, and J. Barlow, "Lifestyle monitoring as a predictive tool in telecare," *Journal of Telemedicine and Telecare*, vol. 13, no. suppl 1, pp. 26–28, 2007.

[24] J. Barlow, D. Singh, S. Bayer, and R. Curry, "A systematic review of the benefits of home telecare for frail elderly people and those with long-term conditions," *Journal of Telemedicine and Telecare*, vol. 13, no. 4, pp. 172–179, 2007.

[25] J. Bardram, T. Hansen, M. Mogensen, and M. Soegaard, "Experiences from real-world deployment of context-aware technologies in a hospital environment," *UbiComp 2006: Ubiquitous Computing*, pp. 369–386, 2006.

[26] C. Orwat, A. Graefe, and T. Faulwasser, "Towards pervasive computing in health care–a literature review," *BMC Medical Informatics and Decision Making*, vol. 8, no. 1, p. 26, 2008.

[27] M. University of Florida and P. C. Laboratory, "Gator tech smart house." http://www.icta.ufl.edu/gt.htm, 2005.

[28] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smart house: A programmable pervasive space," *Computer*, vol. 38, no. 3, pp. 50–60, 2005.

[29] A. Sixsmith and N. Johnson, "A smart sensor to detect the falls of the elderly," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 42–47, 2004.

[30] S. Consolvo, P. Roessler, B. Shelton, A. LaMarca, B. Schilit, and S. Bly, "Technology for care networks of elders," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 22–29, 2004.

[31] I. S. European Commission and Media, "Isisemd project." http://www.isisemd.eu/, 2010.

[32] A. Mitseva, C. Peterson, G. Dafoulas, A. Efthymiou, A. Abildgaard, and S. Bellini, "Isisemd evaluation framework for impact assessment of ict pilot services for elderly with mild dementia, living in the community and their relatives," *Proceedings of the ASTMA NAEC*, pp. 123–148, 2010.

[33] E. program Ambient Assisted Living (AAL), "Rosetta project." http://www.aal-rosetta.eu/, 2012.

[34] M. Morris, J. Lundell, E. Dishman, and B. Needham, "New perspectives on ubiquitous computing from ethnographic study of elders with cognitive decline," in *UbiComp 2003: Ubiquitous Computing*, pp. 227–242, Springer, 2003.

[35] T. Adlam, R. Faulkner, R. Orpwood, K. Jones, J. Macijauskiene, and A. Budraitiene, "The installation and support of internationally distributed equipment for people with dementia," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 3, pp. 253–257, 2004.

[36] A. Mihailidis, J. Boger, T. Craig, and J. Hoey, "The coach prompting system to assist older adults with dementia through handwashing: An efficacy study," *BMC geriatrics*, vol. 8, no. 1, p. 28, 2008.

[37] M. Chan, E. Campo, D. Esteve, *et al.*, "Assessment of elderly mobility using a remote multisensor monitoring system," *Studies in health technology and informatics*, pp. 72–77, 2002.

[38] A. Hauptmann, J. Gao, R. Yan, Y. Qi, J. Yang, and H. Wactlar, "Automated analysis of nursing home observations," *Pervasive Computing, IEEE*, vol. 3, no. 2, pp. 15–21, 2004.

[39] G. Abowd, A. Dey, P. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a better understanding of context and context-awareness," in *Handheld and Ubiquitous Computing*, pp. 304–307, Springer, 1999.

[40] M. Korkea-Aho, "Context-aware applications survey," *Department of Computer Science, Helsinki University of Technology*, 2000.

[41] G. Tsudik, "A family of dunces: Trivial rfid identification and authentication protocols," in *Privacy Enhancing Technologies*, pp. 45–61, Springer, 2007.

[42] M. Conti, R. Pietro, L. Mancini, and A. Spognardi, "Ripp-fs: an rfid identification, privacy preserving protocol with forward secrecy.," in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on*, pp. 229–234, IEEE, 2007.

[43] A. Ranganathan, C. Shankar, and R. Campbell, "Application polymorphism for autonomic ubiquitous computing," *Multiagent and Grid Systems*, vol. 1, no. 2, pp. 109–129, 2005.

[44] M. Musa, M. Mokhtari, B. Ali, M. Rasid, and M. Ghorbel, "Seamless semantic service provisioning mechanism for ambient assisted living," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, pp. 119–125, ACM, 2010.

[45] M. Ghorbel, M. Mokhtari, and S. Renouard, "A distributed approach for assistive service provision in pervasive environment," in *Proceedings of the 4th international workshop on Wireless mobile applications and services on WLAN hotspots*, pp. 91–100, ACM, 2006.

[46] C. Gouin-Vallerand, B. Abdulrazak, S. Giroux, and M. Mokhtari, "A software self-organizing middleware for smart spaces based on fuzzy logic," in *High Performance Computing and Communications (HPCC), 2010 12th IEEE International Conference on*, pp. 138–145, IEEE, 2010.

[47] M. Pathan, K. Taylor, and M. Compton, "Semantics-based plug-and-play configuration of sensor network services," in *Proc. 3rd International Workshop on Semantic Sensor Networks (SSN'10)*, 2010.

[48] D. López-de Ipiña, A. Almeida, U. Aguilera, I. Larizgoitia, X. Laiseca, P. Orduña, A. Barbier, and J. Vazquez, "Dynamic discovery and semantic reasoning for next generation intelligent environments," in *Intelligent Environments, 2008 IET 4th International Conference on*, pp. 1–10, IET, 2008.

[49] S. Helal, "Programming pervasive spaces," *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 84–87, 2005.

[50] T. Erl, *SOA design patterns*. Prentice Hall PTR, 2009.

[51] J. Pérez, J. Álvarez, A. Fernández-Montes, and J. Ortega, “Service-oriented device integration for ubiquitous ambient assisted living environments,” *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, pp. 843–850, 2009.

[52] A. Hein, M. Eichelberg, O. Nee, A. Schulz, A. Helmer, and M. 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*, pp. 531–537, IEEE, 2009.

[53] O. Alliance, “About the osgi service platform, technical whitepaper, revision 4.1,” 2007.

[54] R. Rouvoy, P. Barone, Y. Ding, F. Eliassen, S. Hallsteinsen, J. Lorenzo, A. Mamelli, and U. Scholz, “Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments,” *Software engineering for self-adaptive systems*, pp. 164–182, 2009.

[55] J. M. Marquez, J. Alamo, and J. A. Ortega, “Distributing osgi services: The osiris domain connector,” in *Networked Computing and Advanced Information Management, 2008. NCM’08. Fourth International Conference on*, vol. 1, pp. 341–346, IEEE, 2008.

[56] R.-T. Lin, C.-S. Hsu, T. Y. Chun, and S.-T. Cheng, “Osgi-based smart home architecture for heterogeneous network,” in *Sensing Technology, 2008. ICST 2008. 3rd International Conference on*, pp. 527–532, IEEE, 2008.

[57] H. Bohn, A. Bobek, and F. Golatowski, “Sirena-service infrastructure for real-time embedded networked devices: A service oriented framework for different domains,” in *Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on*, pp. 43–43, IEEE, 2006.

[58] T. Gu, H. K. Pung, and D. Q. Zhang, “Toward an osgi-based infrastructure for context-aware applications,” *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 66–74, 2004.

[59] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Loge, “The smart home concept: our immediate future,” in *E-Learning in Industrial Electronics, 2006 1ST IEEE International Conference on*, pp. 23–28, IEEE, 2006.

[60] F. Jammes, A. Mensch, and H. Smit, “Service-oriented device communications using the devices profile for web services,” in *Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing*, pp. 1–8, ACM, 2005.

[61] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, “Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services,” *Services Computing, IEEE Transactions on*, vol. 3, no. 3, pp. 223–235, 2010.

[62] O. Dohndorf, J. KrUger, H. Krumm, C. Fiehe, A. Litvina, I. Luck, and F. Stewing, “Towards the web of things: Using dpws to bridge isolated osgi platforms,” in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, pp. 720–725, IEEE, 2010.

[63] T. Strang and C. Linnhoff-Popien, “A context modeling survey,” in *Workshop Proceedings*, 2004.

[64] V. Ricquebourg, D. Durand, D. Menga, B. Marine, L. Delahoche, C. Loge, and A.-M. Jolly-Desodt, “Context inferring in the smart home: An swrl approach,” in *Advanced Information Networking and Applications Workshops, 2007, AINAW’07. 21st International Conference on*, vol. 2, pp. 290–295, IEEE, 2007.

[65] M. D. Rodríguez, C. Curlango, and J. P. García-Vázquez, “An agent-based component for identifying elders’ at-home risks through ontologies,” in *3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008*, pp. 168–172, Springer, 2009.

[66] V. Foo Siang Fook, S. C. Tay, M. Jayachandran, J. Biswas, and D. Zhang, “An ontology-based context model in monitoring and handling agitation behavior for persons with dementia,” in *Pervasive Computing and Communications Workshops, 2006. PerCom Workshops 2006. Fourth Annual IEEE International Conference on*, pp. 5–pp, IEEE, 2006.

[67] F. Paganelli and D. Giuli, “An ontology-based context model for home health monitoring and alerting in chronic patient care networks,” in *Advanced Information Network-*

ing and Applications Workshops, 2007, AINAW'07. 21st International Conference on, vol. 2, pp. 838–845, IEEE, 2007.

[68] R. Kadouche, M. Mokhtari, S. Giroux, and B. Abdulrazak, "Semantic approach for modelling an assistive environment using description logic," in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*, pp. 222–231, ACM, 2008.

[69] A. Shehzad, H. Ngo, K. Pham, and S. Lee, "Formal modeling in context aware systems," in *Proceedings of the First International Workshop on Modeling and Retrieval of Context*, Citeseer, 2004.

[70] X. Wang, D. Zhang, T. Gu, and H. Pung, "Ontology based context modeling and reasoning using owl," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pp. 18–22, IEEE, 2004.

[71] G. Klyne, J. Carroll, and B. McBride, "Resource description framework (rdf): Concepts and abstract syntax," *W3C recommendation*, vol. 10, 2004.

[72] D. Beckett and B. McBride, "Rdf/xml syntax specification (revised)," *W3C recommendation*, vol. 10, 2004.

[73] T. Berners-Lee, D. Connolly, and S. Hawke, "Primer: Getting into rdf & semantic web using n3," *World Wide Web Consortium*, p. 16, 2000.

[74] W. Dargie, *Context-aware computing and Self-Managing systems*, vol. 3. Chapman & Hall/CRC, 2009.

[75] T. Tiberghien, M. Mokhtari, H. Aloulou, and J. Biswas, "Semantic reasoning in context-aware assistive environments to support ageing with dementia," *The Semantic Web–ISWC 2012*, pp. 212–227, 2012.

[76] P. Nurmi and P. Floréen, "Reasoning in context-aware systems," *Helsinki Institute for Information Technology, Position paper*, 2004.

[77] J. Helton, "Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty," *Journal of Statistical Computation and Simulation*, vol. 57, no. 1-4, pp. 3–76, 1997.

[78] F. Amirjavid, A. Bouzouane, and B. Bouchard, "Activity modeling under uncertainty by trace of objects in smart homes," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–9, 2012.

[79] F. Amirjavid, A. Bouzouane, B. Bouchard, and K. Bouchard, "Recognition of fuzzy contexts from temporal data under uncertainty case study: Activity recognition in smart homes," in *Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on*, pp. 231–238, IEEE, 2012.

[80] F. M. Ali, S. W. Lee, Z. Bien, and M. Mokhtari, "Combined fuzzy state q-learning algorithm to predict context aware user activity under uncertainty in assistive environment," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD'08. Ninth ACIS International Conference on*, pp. 57–62, IEEE, 2008.

[81] A. Tolstikov, X. Hong, J. Biswas, C. Nugent, L. Chen, and G. Parente, "Comparison of fusion methods based on dst and dbn in human activity recognition," *Journal of Control Theory and Applications*, vol. 9, no. 1, pp. 18–27, 2011.

[82] P. DamiÃ¡n-Reyes, J. Favela, and J. Contreras-Castillo, "Uncertainty management in context-aware applications: Increasing usability and user trust," *Wireless Personal Communications*, vol. 56, pp. 37–53, 2011.

[83] T. Buchholz, A. Küpper, and M. Schiffers, "Quality of context: What it is and why we need it," in *Proceedings of the workshop of the HP OpenView University Association*, vol. 2003, 2003.

[84] K. Sheikh, M. Wegdam, and M. v. Sinderen, "Quality-of-context and its use for protecting privacy in context aware systems," *Journal of Software*, vol. 3, no. 3, pp. 83–93, 2008.

[85] Y. Kim and K. Lee, "A quality measurement method of context information in ubiquitous environments," in *Hybrid Information Technology, 2006. ICHIT'06. International Conference on*, vol. 2, pp. 576–581, IEEE, 2006.

[86] Y. Bu, T. Gu, X. Tao, J. Li, S. Chen, and J. Lu, "Managing quality of context in pervasive computing," in *Quality Software, 2006. QSIC 2006. Sixth International Conference on*, pp. 193–200, IEEE, 2006.

[87] C. Xu and S.-C. Cheung, "Inconsistency detection and resolution for context-aware middleware support," in *ACM SIGSOFT Software Engineering Notes*, vol. 30, pp. 336–345, ACM, 2005.

[88] C. Matheus, M. Kokar, and K. Baclawski, "A core ontology for situation awareness," in *Proceedings of the Sixth International Conference on Information Fusion*, pp. 545–552, 2003.

[89] A. Singh, D. Juneja, and A. Sharma, "A fuzzy integrated ontology model to manage uncertainty in semantic web: The fiom," *International Journal*, vol. 3, 2011.

[90] G. Stoilos, G. Stamou, V. Tzouvaras, J. Pan, and I. Horrocks, "Fuzzy owl: Uncertainty and the semantic web," in *Proc. of the International Workshop on OWL: Experiences and Directions*, vol. 280, 2005.

[91] J. Pan, G. Stoilos, G. Stamou, V. Tzouvaras, and I. Horrocks, "f-swrl: A fuzzy extension of swrl," *Journal on Data Semantics VI*, pp. 28–46, 2006.

[92] T. Gu, H. Pung, and D. Zhang, "A bayesian approach for dealing with uncertain contexts," *Advances in Pervasive Computing*, pp. 136–144, 2004.

[93] R. N. Carvalho, R. Haberlin, P. C. G. Costa, K. B. Laskey, and K. Chang, "Modeling a probabilistic ontology for maritime domain awareness," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pp. 1–8, IEEE, 2011.

[94] A. Bellenger and S. Gatepaille, "Uncertainty in ontologies: Dempster-shafer theory for data fusion applications," *arXiv preprint arXiv:1106.3876*, 2011.

[95] A. Rula, M. Palmonari, A. Harth, S. Stadtmüller, and A. Maurino, "On the diversity and availability of temporal information in linked open data," in *The Semantic Web–ISWC 2012*, pp. 492–507, Springer, 2012.

[96] G. Yang and M. Kifer, "Reasoning about anonymous resources and meta statements on the semantic web," in *Journal on Data Semantics I*, pp. 69–97, Springer, 2003.

[97] C. Matheus, "Using ontology-based rules for situation awareness and information fusion," *W3C Work. on Rule Languages for Interoperability*, 2005.

[98] E. Nazerfard, P. Rashidi, and D. Cook, "Discovering temporal features and relations of activity patterns," in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pp. 1069–1075, IEEE, 2010.

[99] H. Wu, M. Siegel, R. Stiefelhagen, and J. Yang, "Sensor fusion using dempster-shafer theory [for context-aware hci]," in *Instrumentation and Measurement Technology Conference, 2002. IMTC/2002. Proceedings of the 19th IEEE*, vol. 1, pp. 7–12, IEEE, 2002.

[100] H. Byun and K. Cheverst, "Supporting proactive" intelligent" behaviour: the problem of uncertainty," in *Proceedings of the UM03 Workshop on User Modeling for Ubiquitous Computing*, pp. 17–25, Citeseer, 2003.

[101] L. Klein, "Sensor and data fusion concepts and applications," Society of Photo-Optical Instrumentation Engineers (SPIE), 1993.

[102] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[103] L. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy sets and systems*, vol. 100, pp. 9–34, 1999.

[104] S. Parsons and A. Hunter, "A review of uncertainty handling formalisms," *Applications of Uncertainty Formalisms*, pp. 8–37, 1998.

[105] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pp. 57–72, 2008.

[106] A. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 205–247, 1968.

[107] G. Shafer, *A mathematical theory of evidence*, vol. 76. Princeton university press Princeton, 1976.

[108] K. Sentz and S. Ferson, *Combination of evidence in Dempster-Shafer theory.* Citeseer, 2002.

[109] J. Braun, "Dempster-shafer theory and bayesian reasoning in multisensor data fusion," in *PROC SPIE INT SOC OPT ENG*, vol. 4051, pp. 255–266, 2000.

[110] D. J. Patterson, O. Etzioni, D. Fox, and H. Kautz, “Intelligent ubiquitous computing to support alzheimer’s patients: Enabling the cognitively disabled,” in *ADJUNCT PROCEEDINGS*, p. 21, 2002.

[111] J. Boger, P. Poupart, J. Hoey, C. Boutilier, G. Fernie, and A. Mihailidis, “A decision-theoretic approach to task assistance for persons with dementia,” in *International Joint Conference on Artificial Intelligence*, vol. 19, p. 1293, Citeseer, 2005.

[112] P. Olivier, G. Xu, A. Monk, and J. Hoey, “Ambient kitchen: designing situated services using a high fidelity prototyping environment,” in *Proceedings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments*, p. 47, ACM, 2009.

[113] V. Jiménez-Mixco, J. Villalar González, A. Arca, M. Cabrera-Umpierrez, M. Arredondo, P. Manchado, and M. García-Robledo, “Application of virtual reality technologies in rapid development and assessment of ambient assisted living environments,” in *Proceedings of the 1st ACM SIGMM international workshop on Media studies and implementations that help improving access to disabled users*, pp. 7–12, ACM, 2009.

[114] V. Lee, Y. Liu, X. Zhang, C. Phua, K. Sim, J. Zhu, J. Biswas, J. Dong, and M. Mokhtari, “Acarp: auto correct activity recognition rules using process analysis toolkit (pat),” *Impact Analysis of Solutions for Chronic Disease Prevention and Management*, pp. 182–189, 2012.

Index