Davide Rua Carneiro **Simulating and Monitoring Ambient Assisted Living**

UMinho | 2009

Universidade do Minho
Escola de Engenharia

Davide Rua Carneiro

Simulating and Monitoring Ambient Assisted Living

Maio de 2009

Universidade do Minho
Escola de Engenharia

Davide Rua Carneiro

Simulating and Monitoring Ambient Assisted Living

Tese de Mestrado
Mestrado de Informática

Trabalho efectuado sob a orientação do
Professor Doutor Paulo Novais

Maio de 2009

Universidade do Minho, ___/___/______

Assinatura: ____

Acknowledgments

The completion of this work is not only the result of my dedication and love by this research field. Therefore, I am grateful to

Professor Paulo Novais, who has always inspired me to work harder and better,

The VirtualECare team, namely Ricardo Costa and Ângelo Costa, for the ideas shared,

My parents Manuel and Helena and my sister Claudia, who have always supported me,

My girlfriend Elsa for the support, the encouragement and above of all the understanding that attaining a master degree requires time and dedication.

Resumo

A tendência de envelhecimento da população, que se verifica actualmente a nível mundial, começará a ter efeitos em poucos anos. Com um aumento desproporcional da população idosa, a reduzida população activa não será suficiente para suportar os encargos de uma população envelhecida. Para evitar ou minimizar as consequências que daí poderão advir, é necessária uma mudança ao nível das políticas sociais. Mais do que isso, será necessário proporcionar os cuidados de saúde necessários a toda esta população envelhecida para que viva com dignidade e segurança.

É com esta realidade em mente que se apresenta este trabalho, onde se sugere um ambiente inteligente dedicado à monitorização de pessoas necessitadas de cuidados e à prestação de assistência no seu dia-a-dia. A ideia chave é que o utente não seja deslocalizado da sua casa e do ambiente em que sempre viveu e possa, mesmo assim, receber cuidados de saúde especializados e sentir-se seguro. Esta segurança é vivida tanto pelo utente como pela sua família, que sabe que ele está a ser constantemente acompanhado.

Esta dissertação intitulada "Simulating and Monitoring Ambient Assisted Living" surge integrada no projecto VirtualECare. O projecto VirtualECare pretende construir um sistema multi-agente que liga numa mesma plataforma profissionais da área da saúde, pacientes e familiares. Através desta plataforma, o utente será monitorizado, assistido e usufruirá de uma panóplia de serviços que vão desde serviços de saúde especializados a serviços de planeamento, lazer e segurança, entre outros.

No decorrer deste trabalho foi construído um modelo de simulação em que se testou como reagiria o ambiente a determinadas configurações/situações específicas onde se validou o seu funcionamento. Após esta fase, passou-se à implementação do sistema onde se estudou o seu real funcionamento e aptidão perante os objectivos iniciais. O resultado desta dissertação é um sistema multi-agente capaz de monitorizar e assistir os seus utilizadores, providenciando-lhes ao mesmo tempo segurança, sem os retirar do seu ambiente ou em nada alterar a sua habitual rotina, ou a dos seus familiares.

Abstract

The current ageing of the population, visible all over the world, will start to show its effects soon. With a disproportionate increase of the elderly population, the reduced active population will not be enough to support the weight of this new population distribution. To prevent or at least minimize the consequences that may arise, a change is necessary in the social policies. More than that, it is necessary to provide the health care necessary to this group of population so that they can age with the dignity and safety that they deserve.

It is with that objective in mind that this dissertation is presented. On it, an intelligent environment is proposed that is able to monitor and provide care to the user on its day to day. The key idea is that the user should not be delocalized from its house and the environment where he always lived and may, nevertheless, receive specialized health care and feel safe. This security is experienced by both the user and the relatives that know their loved one is being constantly monitored.

This dissertation titled "Simulating and Monitoring Ambient Assisted Living" was developed as a part of the VirtualECare project. This project intends to build a multi-agent system that connects in a joint platform health care professionals, users and their relatives. Thru this platform, the users will be monitored, assisted and will enjoy a wide number of services ranging from health services to planning, leisure, safety, among others.

To develop this work a simulation model was first built on which the behaviour of the environment, given some specific configurations/situations, was tested and validated. After this phase, the system was partially implemented and its real behaviour was studied and tested upon the initial objectives. The result of this thesis is a multi-agent system able of monitoring and assisting a person in need of continued care, while providing at the same time safety, without removing the person from its environment or altering his or his relative's routines.

Glossary

AAL	Ambient Assisted Living
ACL	Agent Communication Language
AI	Artificial Intelligence
AID	Agent Identification
API	Application Program Interface
CAC	Context-aware Computing
CBR	Case Based Reasoning
CCTV	Closed Circuit TV
CNO	Collaborative Networked Organization
DAI	Distributed Artificial Intelligence
EHR	Electronic Health Record
FIPA	Foundation for Intelligent Physical Agents
FIPA ACL	FIPA Agent Communication Language
FSM	Finite State Machine
GDSS	Group Decision Support Systems
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
HSDPA	High-Speed Downlink Packet Access
HVAC	Heating, Ventilation and Air Conditioning
IST	Information Society Technology
ISTAG	IST Advisory Group
IT	Information Technology
JGa	Jade Gateway agent
JVM	Java Virtual Machine
LAN	Local Area Network

MAS	Multi-Agent System
OSGi	Open Services Gateway initiative
PAN	Personal Area Network
PDA	Personal Digital Assistant
RFID	Radio-Frequency Identification
RS	Recommendation System
SOA	Service Oriented Architecture
SQL	Structured Query Language
UMTS	Universal Mobile Telecommunications System
UN	United Nations
UPnP	Universal Plug and Play
VO	Virtual Organization
WAN	Wide Area Network

Contents

List of Figures

1 Introduction

The rapid ageing of the population that is happening all over the world demands for new approaches in the fields of social and health policies. As people get older and active population is not able to support the elderly, social security systems will have to adapt or risk bankruptcy. The solution to this problem may be in the coming together of new technologies that are arising and may provide new solutions for a better and cheaper care providing.

1.1 Motivation

Population ageing is a problem that nowadays affects almost every country in the world. According to its definition [1], population ageing is a shift in the distribution of a country's population towards greater ages. However, on its consequences, population ageing means much more [2]. In most of the cases, it is due to lower birth rates and higher life expectancy. When this happens, the elder population constitutes a bigger percentage of the population distribution. There are evidently both social and economical costs associated. This group of population does not work nor generates richness. Moreover, these are retired people which benefit from the social security systems of their countries. This, along with an increase in health costs, conducts to bigger expenses from the countries budgets. As the situation grows, these economical costs will be too expensive to be supported by the active population, which is diminishing, leading to the bankruptcy of the social security systems. In fact, according to a report prepared by the United Nations [3], not only the working population is declining but also the labour force participation of the older population has declined worldwide over the last decades [4]. According to the same report, the number of older persons has tripled over the last 50 years and will more than triple again over the next 50 years, as we can see on figure 1.1. This is happening all over the globe in both developed and developing countries, being some African and South-American countries the only exceptions. In Portugal, this phenomenon is also visible (Figure 1.2). The main question is if the diminishing active population will be enough for supporting the increasing older population. There is an important indicator on this field: the old-age dependency ratio. This ratio represents the total old population divided by the active population. According to the World Health Organization, in the European Union the old-age dependency ratio was 0.36 in 2002 while in 2025 it is estimated to be around 0.56 [5].

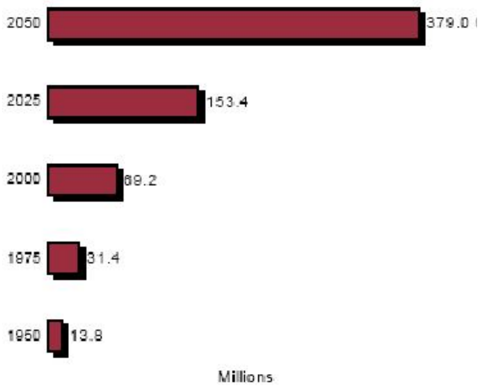

Fig. 1.1: Population aged 80 or over: world, 1950-2050 (source: United Nations [3]).

More than economical costs, this phenomenon has social costs. These people frequently have special needs and require a close and personalized monitoring, mainly due to health related issues [6]. The most common answers are the elder moving to a relative’s house or moving to an elderly home. The first option generally carries major changes to the routine of the host family and not all families are willing to or able to carry that burden. The second one carries increasing economical costs which not every family can support. Moreover, neither is a good option for the elder since they usually show reluctance when being moved out from their habitat where they possibly have their life [7], their friends and their routine. It is mandatory that older people can age and maintain their quality of life while dealing with all the diseases and limitations that arise.

A concept that addresses this subject appeared recently: active ageing. According to the World Health Organization, active ageing is the process of optimizing opportunities for health, participation and security in order to enhance quality of life as people age [5]. It is desirable that while ageing people participate on the society as they always did, the way they can, while ageing. This does not mean that older people continue to work but that they play an important role on their community in social, cultural, civic or spiritual affairs. Older people need, therefore, to feel independent and integrated on their communities, living as they always did, nevertheless having access to the security and health facilities they need.

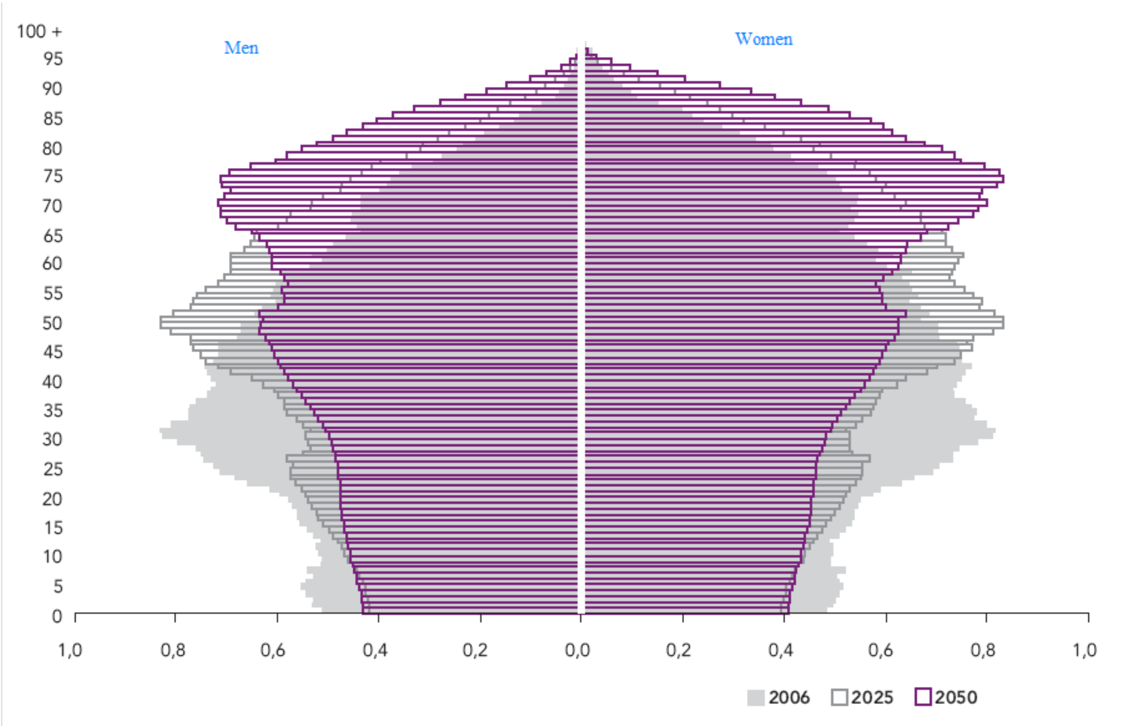

Fig. 1.2: Population pyramid for Portugal: 2006 and projections for 2025 and 2050 (source: Statistics Portugal [8]).

1.2 Scope of the Dissertation

The facts depicted in the previous section raise some challenges that most of the countries must face. There is the need to find new solutions that ensure that all this ageing population will have the deserved care. As health care costs are raising, it is predicted that social security system will be unable to take care of all the ageing population so one of the solutions may be to provide specialized care at home. This could have great advantages since it would keep the elder at home, in its environment, and at the same time take care of him/her. But how to provide this care?

In order to answer to this question, this work proposes a platform able of monitoring both the people and its environment, based on the concept of Ambient Intelligence, more specifically Ambient Assisted Living. This platform should be able to watch over the elder by taking care of its home, adjusting it to its needs or preferences, monitoring its vital signs, reacting every time it is necessary, assisting it in every possible way. By doing so, the elder can live alone, on its environment, having however the safety it needs to maintain its routine as much as possible.

By addressing just a part of the problem one cannot solve the whole problem. Therefore, by being service based, the presented platform is an independent monitoring module that can, however, be part of other higher level architectures, like the VirtualECare project. By doing so, more services can be provided to the persons using it and value is added to the work done.

1.3 Ambient Intelligence

Since the first computers appeared, they have always been seen as one more tool for us humans. Therefore, our interaction with them has always been very computer centred, like with any other tool. When we want to use a computer, we have to go to the computer and interact with it where it is, by its means. This way of interacting with computers is however changing rapidly thanks to new emerging technologies and is leading to what is nowadays known as Ambient Intelligence.

Ambient Intelligence is a relatively new field of Artificial Intelligence. In this paradigm, computers are seen as a proactive tool that assists us in our day to day to make our lives easier. We no longer need to interact with computers in ways that are not natural for us since in this paradigm, a main concern is that computers can interact with us using friendly interfaces, like our natural language or gestures. Intelligent Environments also tend to be as invisible as possible as computational power and communication technologies are nowadays present in almost every device we use. The trend is therefore that one interacts with one or more computers without probably even noticing it. The only thing we should be aware of when interacting with an Intelligent Environment is of the results of that interaction: more safety, comfort and well being in our day to day.

1.4 The VirtualECare Project

The VirtualECare Project [9, 10] main objective is to present an intelligent multi-agent system able to monitor, interact and provide its customers with health care services of the utmost quality [11]. This system will be interconnected, not only to other healthcare institutions, but also with leisure centres, training facilities, shops and patient relatives, just to name a few.

The VirtualECare Architecture is a distributed one with their different modules interconnected through a network (e.g. LAN, MAN, WAN), each one with a different role (Figure 1.3), a top-level description of the architecture machinery is given:

- Supported User – elderly people with special health care needs, whose clinical data is sent to the CallCareCenter and redirected to the Group Decision Support System. This user should be constantly monitored, outside and outside its environment so the data must be provided in real time to the interested parts. It is the central component of the architecture and all the other components must work together to ensure its safety and well being;
- Environment – the elderly natural environment, provided with sensors, with the clinical data being sent to the Group Decision Support System through the CallCareCenter, with the remaining ones being redirected to the CallServiceCenter. The data provided by this module must also be constantly available and analyzed so a reliable network connection is mandatory. The environment can be the user home, a hospital room, a day centre, just to name a few. The main actions of the other components towards the environment are to maintain the comfort and security parameters;
- Group Decision – This module is responsible for the long term planning regarding the health care of the patients. It should be composed of specialized staff like nurses and doctors as well as Recommendation Systems and tools for time and space distant meetings. In the overall this module should be able of planning all the issues related to visits to doctors, tests, automatically scheduling all this according to the user agenda;
- CallServiceCenter – Entity with all the necessary computational and qualified personal resources, capable of receiving and analyze the diverse data and take the necessary actions according to it;
- CallCareCenter – Entity in charge of computational and qualified personal resources (i.e. healthcare professionals and auxiliary), capable of receiving and analyze the clinical data, and take the necessary actions according to it. The user may make voice calls to this service and

request assistance or advises and the service should respond and, if necessary, contact other modules like the Group Decision;

- Relatives – individuals that may have an active role in the supervising task of their love ones, being able to give precious complementary information about them and being able to intervene, in a complementary way, in specific crises (e.g., loneliness). By being an important part of the equation, the relatives should also have access to the health status of the patient so that they are constantly aware of its situation.

In order to the Group Decision Support System to make its work, it has to collect the opinion of specialized staff (e.g., nurses, paediatrics, cardiologists). There is also the need to have a digital profile of the Supported User, allowing a better understanding of his/her special needs. In this profile we can have several different kinds of relevant information, ranging from the patient Electronic Clinic Process to their own personal preferences (e.g. musical, gastronomic) passing by their own personal experiences, which can be used to better understand and satisfy their needs and expectative.

This solution will help healthcare providers to integrate, analyze, and manage complex and disparate clinical, research and administrative knowledge. It will provide tools and methodologies for creating an information-on-demand environment that can improve quality-of-living, safety, and quality of patient care.

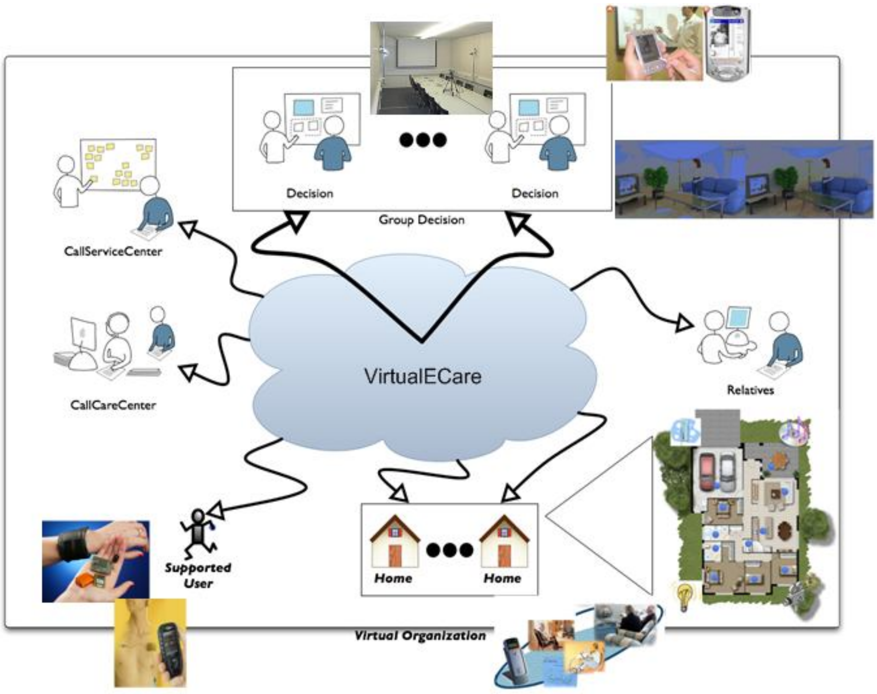

Fig. 1.3:The VirtualECare architecture.

1.5 Related Projects

In the following sections some projects that were analyzed during this work are presented. All of them cover complementary areas and all of them have benefits for users in assisted environments. They are the evidence that many other researchers believe that new technologies are the solution to this problem.

Amigo – Ambient Intelligence for the networked home environment

This project was a consortium of fifteen European companies who joined efforts to exploit the potential of the nowadays common home networks and improve people's lives. The idea was to take profit of the fact that nowadays almost every equipment comes with a network connection. But still, there are lots of different standards which make more difficult their interoperability so they wanted to make that possible. This project ended in September 2004.

It is common that a house has several networks such as the electrical, Ethernet or wireless networks. The Amigo project [12] interconnects these networks thus enabling communication between all devices. Over this "Hardware" layer, Amigo implements services so that people's environments are empowered. From any point of the house or even from outside the house, people can change house parameters, watch the surveillance cameras, put their TV to record some program, etc. This is in fact the main purpose of the project, to empower the environment it is in, releasing people from boring activities as the person is the center of the system. This project used 'home laboratories' across Europe including Philips Research's 'HomeLab', France Telecom's 'Creative Studio Lab' and the Fraunhofer Institute's 'InHaus'.

Oxygen

This project from MIT [13] has as main objective to make technology as available everywhere as the air we breathe. Since the beginning, computers are closed in rooms, and we have to get to them, interact with them using their means, work with them the way they want. Meaning that until now, we have been living for the computers instead of the computers living for us. MIT vision is that in the future, "computers" are available everywhere, so that everyone can use them and we don't need to carry our own devices. "Computers" will also be very generic, configurable to fit all our needs in every moment.

In order to do that, this project wants to make interactions between person and computer as natural as possible, using people's natural language or gestures. Computers will live for us, expanding our possibilities, simplifying our lives. They will learn our preferences, learn what we like, providing an even more natural interaction. They will be everywhere, watching our safety, taking automatically care of our needs, fetching information before we asking for it.

Several modules necessary for such a project to work have been developed. The Intelligent Room has a speech recognition system which receives and executes orders from people in the room. Imagine you are in a meeting, you can ask the room to read what is scheduled for today, show some video of the last meeting, show or read some document, etc. Another interesting technology is the Cricket/INS. This is a people location system inside a building that uses a badge carried by the people being traced. The main advance here is that the services someone is using follow the person as he or she travels from a room to another. Imagine you are listening to some music in your bedroom. When you walk into the living room, the system locates at each time where the nearest speakers are and automatically the sound starts playing there and stops playing in the bedroom. The same is possible for lights, air conditioning, TV or any other service.

I.L.S.A. – The Independent Lifestyle Assistant

This initiative from University of Minnesota [14] has as main objective to study the response of elderly to a monitoring computer system inside their houses and determine how such systems can help this people. They not only determined which are the main problems of elder people living alone, but they also implemented parts of a monitoring system in some test houses in real conditions. This application of AmI is not only good for the elder living alone but also for their caregivers as everyone maintains its autonomy.

A group of sensors was placed in each house (eleven houses during half year) according to what was being monitored (behavioural patterns, medication taken, etc). The information from the sensors was read and sent to a central where it was studied. From this central, alerts where emitted if the person in question did not take the medication or if the behavior during the day was very unusual. The main features implemlented included passive monitoring (mobility, occupancy, sleeping patterns), cognitive support (like reminders), alerts and notifications, reports (summary reports of client behaviour) and controlling remote access to information. Clients had a portable device from where they could check their agenda, change some system parameters and even communicate with caregivers. This is a project that directly interacted with a specific public:

elder people living alone. This public of course has its own needs and creates specific problems or challenges that must be addressed and that was the target of this project.

Intelligent Transportation System

Intelligent Transportation System (ITS) is an initiative that intends to join information and communication technologies with the transportation system and the vehicles on it. It was created after the boom of vehicles in our roads as a way of fighting problems such as congestion, pollution or fuel consumption.

It comprises several technologies from simple GPS or traffic light control systems to CCTV circuits and Parking Guidance and information, all interconnected wirelessly. This would be a "simple" monitoring system if it didn't have some predictive capabilities. In fact, using additional information like weather forecasts or current congestion level, it is now possible to predict future congestions for example. The system can also communicate with our car navigation device and inform it of some accident ahead and change our course. Hence, by detecting congestion and automatically deviate traffic through less congested roads, the system improves the efficiency of the transportation system. As every car is connected and visible to the traffic system, if an emergency vehicle approaches a traffic light that is red, the ITS can control the light signs so that the ambulance can continue safely.

These kind of systems are now being implemented and have already some history, namely in the USA and in Europe. ITS is by itself an amazing project with huge possibilities that can be even more expanded when integrated with other AmI projects.

ReachMedia

This project from MIT [15] consists on an RFID equipped wristband to provide us with on-the-move interaction with everyday objects. Usually, there is a lot of information related to the objects we deal with every day, mainly on Internet but it is normally only accessible through a computer. This project aims to present us that information wherever we are, in real time, without driving our attention from what we are doing.

The wristband contains an RFID reader that will read the information from RFID tags in objects close to our hands. After this, information is fetched from the Internet and presented to the user in some interface. At this moment a phone is being used to fetch the information and the user listens to what is found. Imagine you are in a book store and you grabbed some book you want some information about. When your hand approaches the book's RFID tag, the system beeps notifying you that some services are available for that object. For a book, there can be reviews or ratings that some store or book specialist put on-line. You can then choose, while you are flipping through the book's pages, what you want to listen about that book.

The navigation between the several choices is also done using the wristband. As it is equipped with accelerometers, with small gestures of the wrist, the person can navigate through the several options for the object in question and select what to listen to. The uses for such a technology are wide. When meeting people with the same wrist, we could know what their interests are, their hobbies or what their personality is like. When shopping, we could know the characteristics of every product we grab (e.g. calories of alimentary products) while we are walking and looking at other products instead of having to stop and read the product specification.

Telecare

The objective of the Telecare [16] project is to develop a configurable framework for assisting elder people, based on the integration of a multi-agent and a federated information management approach. The result are services likely to be offered by the emerging ubiquitous computing and intelligent home appliances, which are useful for elderly people. With this approach, the project expects to address issues like elderly people being moved from their homes, providing them with autonomy and independence. To achieve these objectives, this project is based on tele-supervision and tele-assistance technologies. A virtual network is created, which connects the elderly home, the relative's office, the care or leisure centres, a virtual shop among others. It is thru this virtual community that the elderly makes use of the services. The project states that it is possible nowadays to create such a network that can provide cheap health care to elderly, namely because of the current development of internet-based infrastructures. The development of such projects is one important step towards countering the problems of ageing population and possible elderly marginalization.

1.6 Simulating and Monitoring Ambient Assisted Living

As population ages and social problems grow, new solutions for this problem must be devised. The providing of care at home may be a solution that benefits both patients and health services providers. While maintaining the patient at home, in its environment, the costs can be diminished and at the same time the comfort and well being of the person in need of care increased. To pursue this goal, one can rely on the telecommunication technologies but this, by itself, is not enough. The people in need of care need to feel that there is actually something taking care of its needs, watching over its health status constantly and not only when the person decides to call the doctor. The answer to this challenge can be the Ambient Assisted Living environments.

Ambient Assisted Living projects can be helpful to people in a wide range of domains: in a work environment to assist workers on their tasks, in a medical environment to support staff and patients or also in a home environment, to assist people in their day to day living. This multitude of application possibilities and the growing interest in Ambient Assisted Living are visible in the scientific events dedicated to this field that have been appearing in the last years. As examples, we can mention the Symposium of Ubiquitous Computing and Ambient Intelligence (UCAmI)[1], the International Workshop of Ambient Assisted Living (IWAAL)[2], the European Conference on Ambient Intelligence[3] or the Journal of Ambient Intelligence and Smart Environments (JAISE)[4].

In the specific scope of this work, these environments are going to be studied for the advantages they can bring to people in a home environment, narrowing down to the problematic of the health care services at home. We believe that it is possible to provide continuous health care services and, at the same time, increase the comfort, well being and safety of a person while maintaining her on its environment. In the end it is expected that this work contributed as much as possible to making this a reality.

[1] The website of UCAmI is available at http://ucami.usal.es/ <Accessed on May, 2009>.

[2] The website of IWAAL is available at http://gsii.usal.es/~iwaal/ <Accessed on May, 2009>.

[3] The website of the European Conference on Ambient Intelligence is available at http://www.ami-09.org/ <Accessed on May, 2009>.

[4] The website of JAISE is available at http://www.iospress.nl/loadtop/load.php?isbn=18761364 <Accessed on May, 2009>.

1.6.1 Technological Challenge

An Ambient Assisted Living environment can be made of many different components. These components can be small handheld devices, sensors, home appliances or software applications just no name a few. Moreover, each device can use different technologies which raise the first main challenge to address: how to integrate in the same architecture a multitude of different devices and technologies. There is the need to find a backbone that can support so much and so different components. Also, there is the need for a communication mechanism that all of the components can use so that each one of them can share information with the others, maximizing the possibilities of the system.

When having these much heterogeneity in an architecture another challenge raises: the development of applications. Each component has its specific functionalities which it enrich the environment. However, if the developer wants to create an application that uses some or even all of the components of the environment, he must be a specialist in each one of the components and know how to interact with the device and how to use its functionalities. This evidently increases the development time and the complexity of any application developed.

Because of the great number of devices that can exist, we must also think on the problem of merging them with the environment. An intelligent environment should appear a perfectly normal one, showing only the common devices or appliances. This also deals directly with the acceptance of these kinds of environments. People tend to accept better such systems if they seem to change nothing in their lives, mainly when we talk of elder people. If such environments bring many changes to their homes, with lots of new devices that they probably never heard of, the acceptance will diminish and even if the system is a very good one, it may fail.

An architecture like that must also be dynamic. One does not expect that with the rate at which new devices appear there will not be more devices added to the architecture. The user can even, just because he wants to, remove a device or replace it for a new one, with different functionalities and probably with a new technology. This is also a challenge that must be addressed. The system will not be static and devices that were available at the beginning may not be available at the moment as well as it is desirable to search constantly for new devices that can provide new helpful services or the same services in a more effective way.

It is also desirable that such a project is not a closed one, enabling interconnection with similar systems so that functionalities and advantages can be shared through both systems and users. With so much complementary projects appearing every year, the compatibility and interoperability between them can be a major advantage, mainly for the users since one problem we face nowadays is that there are many devices and systems meant to assist people but each one works on its own, with its rules, which makes it difficult to put them to work together.

At last on may talk about the challenge itself that is to build such a heterogeneous architecture. There are many variables that cannot be predicted and many risks. As there is the need for acquiring many devices with new technologies, there is the risk that those devices reveal themselves unfit for the task, whether for its technology or for its functionalities. There is the need to test and study some ideas so that when they are accomplished there is a minimal certainty that they will be feasible.

1.6.2 Objectives

The work documented in this dissertation was developed under the VirtualECare project. In the context of the high level architecture presented before, this work focuses on two of the main modules: the Supported User and the Environment, which also addresses the external environment. We can therefore now look at a new high level image of the architecture, which highlights the components developed and documented in this dissertation (Figure 1.4).

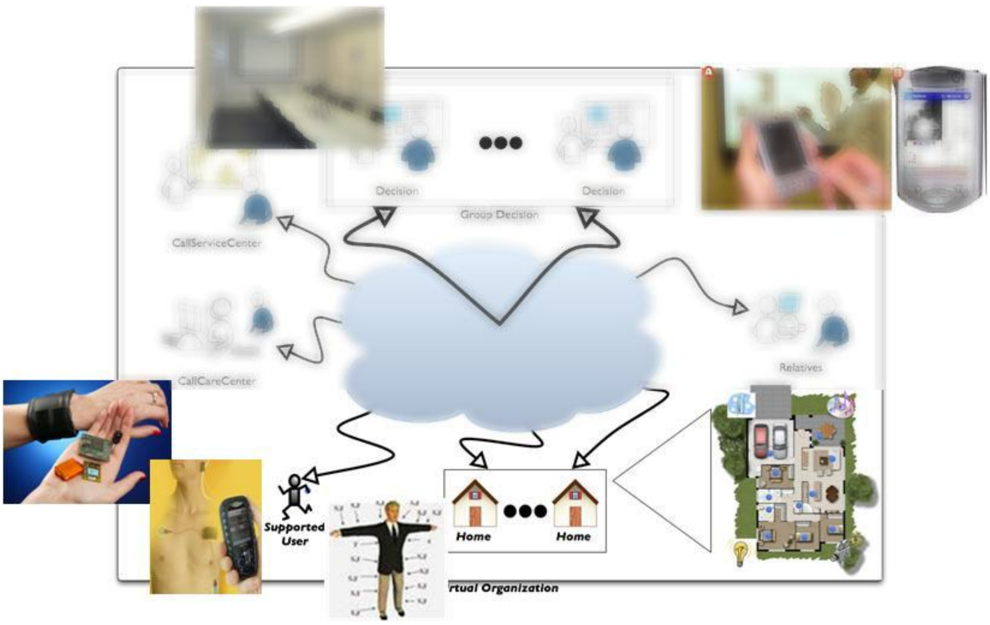

Fig. 1.4: The part of the VirtualECare architecture focused in this work.

The main objective of this work is to create a system that can provide some basic care and well being for persons in their home environment, with the aim of fighting the social problems depicted before, merging already existing and available technologies. In order for this work to be accomplished, the following objectives were set:

- Develop an architecture for Ambient Assisted Living. The architecture must respect the following features:
    - Support a large number of heterogeneous components and their intercommunication;
    - Provide a layer of abstraction that hides all the singularities of each device and enables an easy development of applications that use the devices in the environment;
    - Use common devices and merge them into the normal environment, taking profit of their new computational and networking abilities;
    - Be dynamic, being ready to adapt in time to different situations as they occur (as the addition of new components);
    - Be easy to integrate with other architectures, using other's services and providing its own so that advantages can be shared.
- Design a simulation platform that allow for the system to be tested and its effectiveness improved before acquiring all the equipment. This is important for validating the architecture and the protocols used and give us more reliability on the final version;
- Implement the architecture defined and improved;
- Enrich the architecture with the ability to learn from the user how to manage the environment.

1.7 Investigation Methodology

To accomplish the objectives enumerated before the Action-Research methodology was followed [17]. This methodology starts by identifying the problem so that a hypothesis can be formulated on which the development will be based. Subsequently, the information is recompiled, organized and analyzed, continuously building a proposal for solving the identified problem. Finally, one can make its conclusions based on the results obtained during the investigation. For this research model to be followed, six complementary stages have been defined to achieve the planned objectives. The stages defined are described ahead:

- Specification of the problem and its characteristics.
- Constant and incremental update and review of the state of the art.
- Idealization and gradual and interactive development of the proposed model.
- Experimentation and implementation of the solution thru the development of a prototype.
- Result analysis and formulation of conclusions.
- Constant diffusion of knowledge, results obtained and experiences with the scientific community.

1.8 Structure of the document

This document is organized as follows: Chapter 2 contains a detailed description of Ambient Intelligence, which starts by emphasizing its main characteristics and showing the high level architecture. It continues by detailing some important components of an Intelligent Environment like sensors or actuators, and the advantages of having such environments working for Humans and exposes the context of service providing from the AmI point of view. It concludes by showing some common and useful applications of AmI, particularly in the Health Care Sector and the expected future of Ambient Intelligence.

In Chapter 3 the concept of Ambient Assisted Living is introduced in terms of the main paradigms behind it and its relation with e-Health.

Chapter 4 details the architecture needed for having an Ambient Assisted Living environment. It starts with a comprehensive description of the technologies most used, continues enumerating devices that can be used for implementing these kinds of environments and the bridge between the system and the person and finishes by specifying an architecture for supporting Ambient Assisted Living.

In Chapter 5, the simulation platform that was created during this work is detailed. It emphasizes the importance of simulation tools, it details how environment, sensors, actuators and users are simulated and how the simulation shapes the final architecture, acting as a test bench. It also shows some screenshots of the simulation tool running.

The Monitorization platform and its components are detailed in Chapter 6. It is described in terms of the sensors and actuators that were used, inside and outside the environment. It also explains how the service description was achieved so that a computer system can analyze each service and decide what is the best one to use when needed. It ends by describing the methodology used by the system to learn from the user how to select the services to provide and how to learn its preferences.

The document ends with Chapter 7, where a Synthesis of the work done is presented, as well as some important contributions and work generated. It finishes with the conclusions obtained and the future work that has yet to be done.

2 Ambient Intelligence

Ambient Intelligence is the main concept behind this thesis and present in all the work done. This section starts with a detailed description of AmI, it details the main components behind it, its main uses and the predicted future.

2.1 What is Ambient Intelligence?

Ambient Intelligence [18, 19], as defined by the IST Advisory Group (ISTAG) is a relatively new paradigm, born thanks to three new key technologies: Ubiquitous Computing [20], Ubiquitous Communication [21] and Intelligent User Interfaces [22] which is starting to change the way we see computers (Figure 2.1). It may be, in fact, the first time computers will start to actually work for us, instead of us working with them. Until now, computers have been more like a tool, doing the tasks we program them to do. Nothing distinguishes them from a hammer or any other tool except for the fact that computers can be programmed to do different tasks while hammers can only hit stuff. Now imagine a hammer which hits the stuff we want, when we want, how we want, without us having to grab and balance it. That is what is happening in AmI. Computers, in intelligent environments, are no longer mere tools but are learning what we like, what we do, our habits and our preferences so they can simplify our lives. Moreover, they are shrinking and hiding in common devices so that we don't even notice them.

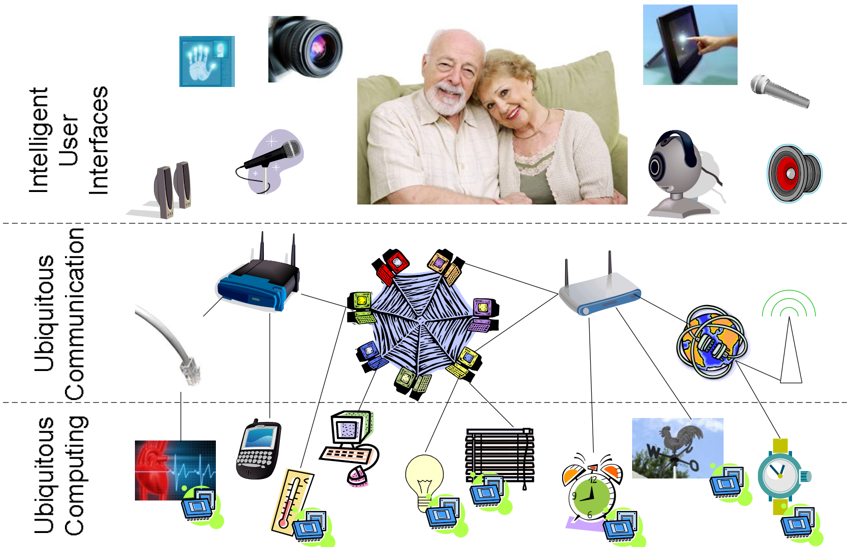

Fig. 2.1: The three main layers of Ambient Intelligence

Physically, an intelligent environment is composed by the ambient itself (can be a house, a room, a car, a school, an office...) and the devices on it. These devices are common devices like our mobile phone, air conditioning systems, a laptop or desktop, media servers, micro-waves or PDA's, all of them so common these days. What these devices have new is that they are all connected through a control network so that they can be controlled or control other devices from any point of the network. But this, by itself, would only be what be nowadays call domotics[5]. In domotics, devices are connected so that we can control them from distance but that isn't actually AmI. AmI goes further as we will see in this section.

So, what makes AmI more than just a group of components connected together? First of all, it's the fact already mentioned that the components are as hidden as possible. The ideal intelligent environment should appear a perfectly normal environment, embedding its components in common devices, noticeable only by its actions. This is why AmI depends on UbiComp: to try to integrate computational power into small devices, so that they can pass unnoticed but, nevertheless, do their job. This distributed architecture makes agent technology very fit to implement intelligent environments, as we will see ahead. But what is their job?

Their job is to ensure people's well-being and safety. In order to do that, they must know the needs and the preferences of the person. Preferences and needs can be set manually, when configuring the system. Moreover, the routine and the preferences can be learned as the person

[5] Domotics or Home Automation is the automation of private homes with the objective of providing security and comfort to its inhabitants.

does its day-to-day [23]. This is a characteristic of AmI, it *learns* just by interacting with the user, without him even noticing, the system is studying his behaviour, learning what he uses to do and when he uses to do it. With learning, another aspect comes along. If the system learns the habits of the person it interacts with, it means that every environment will be unique, depending on the persons who interact with the system. So we can say that another important characteristic of AmI is that it is *personalized* [24]. Moreover, we can also say that it is *adaptive* since, even after learning our habits, if we change what we used to do, the system will adapt and learn the new habits, it will adapt to us and to our lifestyle [25].

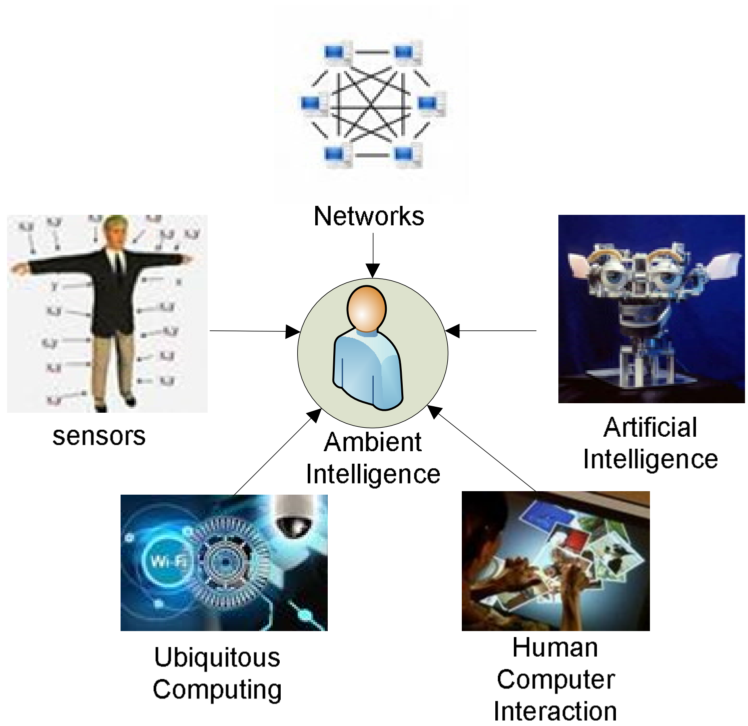

Fig. 2.2: The main areas of Computer Science behind AmI.

But, what is it good for to learn the user habits if we don't take advantage of it? And what would be the best way of using such information about user habits or routines? The answer is simple. After learning how the user behaves, the next thing to do is to predict what the user will do and take measures that assist the user in that action. So, if every day the person drinks a coffee when waking up, the system learns and turns on the coffee machine when the alarm clock sounds. So, Ambient Intelligence can also be described as *proactive*.

In order to correctly choose how and when to provide some service, the system must be aware of the context of the person. The context, in AmI, is described by the action some person is performing and where it is being performed, what is the state of the environment at that time, what is the emotional and physical state of the person and eventually other factors like the weather or the traffic depending on the location of the person. This means that AmI must have methods for describing the current situation of the person and the environment around. Therefore, AmI is also described as *context-aware* [26].

In a few words, AmI can be defined as “A digital environment that proactively, but sensibly, supports people in their daily lives“[27]. This sensibility comes from the intelligence, much like in real life: a trained nurse that can identify symptoms can proactively provide better care. If the nurse knows the user and its preferences or needs, she can be sensible towards the user. In order for AmI to show these characteristics, it makes use of several fields in Computer Science, being the most notorious: Artificial Intelligence, Human Computer Interaction, Sensors, Networks and Ubiquitous Computing (Figure 2.2).

2.2 Ambient Intelligence as Service Providing

AmI can, in a way, be seen as the continuous providing of services[6] to the users. For example, a teacher, when entering a classroom, might have as services the control of the lights, the air conditioning, a digital assistant or an interactive whiteboard. A student, when entering the same room, might not have access to the same services. With this in mind we can identify some important propositions for an intelligent environment to work.

First of all, the system must have as much information about the users in the environment as possible. It must identify the user and know its role in the environment. It should also know the limitations or preferences of that user. We can think of an impaired user which cannot reach a certain object. There is no use on the system telling the user where the object he needs is if the user cannot reach it. As for the preferences, it is very important to try to meet the preferences of the user when one of the objectives of Ambient Intelligence is to provide well being. Both this cases indicate that an Intelligent Environment should adapt its services to the users. This means that the services cannot be static, rather they should be personalized according to who is requesting them.

Having all this information about the users, the system will decide which services to provide, when to provide and how to provide. The system must also know the availability of the

[6] A service, in the ambit of this document, is defined as something (physical or not) that is available in the context of the user, that performs some useful task and can be used by him when needed.

devices eventually needed for providing the service. It should not announce a service if it is not possible to provide it. Other important characteristics about the services are the geographic location, the costs, the probable effects, among others so that when it suggests some service to the user, it suggests what it thinks to be the best service. In other words, the system must be context-aware relatively to the users and their environment and have a strong description of the services so that the user well being is maximized.

The subject of adapting the services to specifically meet the needs of elderly people [28, 29] is in fact a very important one. This group of people usually has physical impairments like mobility limitations, partial or total blindness or deafness and other issues that prevent them to use the appliances or devices in a regular way. As said before, services in AmI must be adapted to meet the needs of the users, mainly when the users have these particular problems. This is, by itself, a very interesting field of investigation and lots of work can be done here. This means that if the services are correctly adapted, the elder or impaired population are a class of persons that have much to profit with Ambient Intelligence since it can empower their lives giving back the autonomy they had lost.

2.3 Sensing

We have, until now, described some of the main features of AmI. We said that it learns our behaviours, it adapts to us, it hides in common devices, it is aware of our context and it is able of predicting our actions. Let's now talk of what is behind AmI, what are the devices upon which this behaviour can be achieved.

One thing AmI really depends on is sensors. A large sensorial network is indispensable for correctly reading as much data as possible from the environment and from the user, as this data is the basis of any decision from the system. So, we can connect for example the nowadays common smoke detectors, flood detectors, gas detectors or intrusion alarms. This would take care of the safety in the environment. Adding sensors for luminosity, temperature or humidity, makes it possible to read the environmental parameters so that the environment can be monitored in order to maintain the preferences, the needs or the safety of the person. The person location inside the house is a more complex subject that can be addressed through the use of several types of sensor since one, by itself, is not enough for such a task. For determining in which division the person is, AmI can take profit of motion detectors or technologies such as RFID. For determining where exactly the person is inside the division, it is possible to use weight sensors on furniture like sofas or beds or even detect the activity in some devices: if the person just turned the coffee machine on, this means that it is close to the machine. More complex location systems can also be built using triangulation and RF technology.

Another very important use for sensors in Ambient Intelligence is for monitoring the user vital signs. There are nowadays many accessible ways of doing it. There are jackets, wristwatches and other implementations of what is called wearable computing [30]: using clothes with embedded computational power. In this specific case, the user can wear clothes that have sensing capabilities. These clothes then share the information read from the sensors with the system using wireless protocols, mostly Bluetooth.

There are much more types of sensors that can be used an enrich the information that a system can acquire from the environment. A final example can be outside weather stations which, when connected to the system provide information about the exterior environment. This may be important not only to inform the user but also to make recommendations regarding what to wear or what to do this day, according to the weather state.

2.4 Acting

In this section we will talk about the result of the work of an Intelligent Environment: the actions. These actions are mainly observed on the environment itself but they may be also directed to the users of the environment or even other external entities. Generally, the Intelligent Environment can take an action due to two main causes: a reaction to an event or an action to prevent or cause an event. To these two cases we call reactive and proactive behaviours, respectively.

In the first case, the system detects a predetermined event and reacts to it in a predetermined way. We can think on the lights being turned on when the client enters the house or the heat being turned to minimum when the client is sleeping. In the second case, the system, by its initiative, takes an action. In this case, we can think of the system turning on the coffee machine at the usual hour the person wakes up or turning on the heat at home at the usual hour the person arrives from work.

The actions can yet be classified according to its purpose. The system can take an action concerning the safety of the user, concerning its well being or simply for assisting him. In these three cases, we can think of the fire-fighters being called automatically after a fire alarm, in controlling the heat and in turning on the coffee machine for him as examples.

The most important actions are obviously the ones that concern the user safety. In this area, an Intelligent Environment can constantly monitor the state of the environment and rapidly react in case of danger, even if the user is not at home. If there is a fire, a flood, or gas is detected inside the house, an Intelligent Environment has enough autonomy to automatically call for help and prevent the situation to get worst, without interacting with any user. When talking about the user's vital signs it is even possible to detect a heart attack before it happens gaining precious seconds in favour of the user.

The actions that concern the user comfort are probably the ones more used in an Intelligent Environment. They are mainly intended to interact with all the components in a house that can change the environmental parameters. In this group of components one can think of air conditioning systems, heaters, dehumidifiers and even windows, doors and blinds. The objective is for the system to be able to change the environmental parameters thru the use of these devices so that the user preferences can be met. Of course that when acting this way, the objective can also be to save energy. In an Intelligent Environment, it is even possible to increase or maintain the comfort of the environment inhabitants while at the same time diminishing the energy bill [31].

At last, the actions that have has objective to assist the user. These actions can be helpful for us in many different types of environments. Imagine that your workplace is enriched with an Intelligent Environment. It is constantly examining your work and identifying what you are doing so that it can proactively help you. The way it helps depends on the nature of the work but it may be by taking care of your daily schedule if you are a nurse, by showing you a schematic of the electric device you are currently putting together or by automatically fetching components needed for finishing assembling a car on the assembly line. In the specific scope of this work, these actions can be even more profitable for elder or impaired users. Imagine a user that cannot reach a device on its house because it is on a wheelchair or it simply is too far to walk to it. An Intelligent Environment would respond to a voice command or even automatically turn on that device, this way assisting the user.

2.5 Communicating

All the devices mentioned in the previous sections have to be connected since they need to share information. Several communication technologies suitable for use in AmI are explained in detail further ahead however, we must introduce here the importance of the network in the architecture of any intelligent environment. The wired network is very important in connecting devices with low mobility and is very common nowadays, working also as a backbone for other networks. We have the example of the widely used Ethernet protocol, which uses a dedicated line for data transmission, or the less common Powerline, which uses the electric signal of the power line as a carrier of data, while, at the same time, providing power to the devices. It is common to have wireless networks supported by these wired networks, so that portable devices can also integrate the network. Moreover there should also be a gateway through which devices and users inside the environment access the outside network. Regarding the sensorial network, there are several protocols for connecting sensors which are oriented for low power consumption and low bandwidth, since sensors don't need to transmit much data. The result of all these different protocols working together is a heterogeneous network which allows very different types of devices to be connected.

Because of the existence of so much different devices communicating, there must be a common language to be spoken between all of them: we don't need just the air so that our voice can go from the speaker to the listener, we also need to share the same language so that we can communicate. The same happens here. Different devices use different infrastructures and different protocols for communicating and there must be a way for making the bridge between the different protocols. There are some possible solutions for fulfilling this goal. One of the most promising ones relies on the Service Oriented Architecture [32] paradigm. In this paradigm, what is visible is not the device itself but rather what the device can do, what the device has to offer. It does not matter how the device works or what it is, it only matters what it does. Services are announced in a common language that every device knows and these same services can be used by every device in the architecture. This may be a solution for integrating the heterogeneity that is visible nowadays with the emerging of networking devices.

2.6 Using

Another important component of an intelligent environment is the interface with the person. Interfaces have never been very user friendly. For years, one had to interact with computers first thru a keyboard, later also with a mouse pointer but that has not changed much. However, that is not the most intuitive way, especially for people that are not used to it or have poor contact with computers. Interfaces in AmI tend to be much more user friendly so that everyone can use them, despite their experience with computers. This fact is even more important when it comes to impaired persons, which can have a big difficulty or even be unable to use the common interfaces that we use.

This way, interfaces can be built upon very different devices, depending on the type of environment and the target population. In a classroom for example, the interface could be an interactive board, while in a house could be a touch screen or a handheld device. The trend is that interfaces get as natural for us humans as possible. Therefore, it is expected that in the future our hands and our body will no longer be stuck to the computer but our interaction will be with our voice, in our natural language and with gestures. This way, we can be doing other things and, nevertheless, interact with the computer, anywhere.

Another important characteristic about interfaces that sometimes is ignored is that the interface is the visible face of the system. One can have an amazing Intelligent Environment but if its interface does not explains to us what it does, if it is not friendly and easy to use, if we feel that we have no control, the acceptance of the system is bad. The interface should also adapt to the type of user it is dealing with. Maybe to an older person it would be better to keep the interface simple, giving less information and in a more explicit way while to a younger user which has a better understanding about computer systems the system could provide more information and in a more rapid way. Interfaces are therefore a very important part of any intelligent environment.

2.7 Enjoying AmI

Now that we know what AmI is and the main components behind it, let's see what this paradigm can do for us and what can we profit from it. As said before, an intelligent environment is an extension of a conventional environment, trying to appear as traditional as possible, but assisting us in every way, making our life easier. In this aspect, AmI can be classified in three types, according to the way it behaves towards us: decision-support, control-support and assistive, each one of them with its own features.

A decision-support environment would help the users decide within some possible choices, or would itself generate some possible decisions or scenarios given the user actual context and providing the pros and cons. A simple example with home application would be generating one or more recipes, based on the goods available on the refrigerator. A more complex example can be found in [33] where a decision-support environment helps an elder decide what to do on weekend, based on factors like the weather, his health status and if there are relatives visiting him or not.

A control-support AmI is more oriented to help the user with the control of the environment itself. Already common yet simple examples of control-support AmI found in homes or offices are systems that control the presence of people in the rooms for energy saving and personal preferences satisfaction (by turning off the lights in unoccupied rooms, by maintaining the temperature the person likes, etc). One of the major uses for control-support AmI already widely used is the control of HVAC in big buildings like skyscrapers, shopping malls or big factories. HVAC is the acronym that stands for Heating Ventilation and Air Conditioning and it refers to the modern systems we have nowadays that control every environmental parameter inside buildings. In standard buildings the objective is to ensure the comfort of the persons living or working there. More specific buildings require more specific parameters. A building like a library or that hosts a server farm would be interested in maintaining the humidity as low as possible while in hospitals, these systems spare are also interested in maintaining a constant temperature and constantly renewing the air. No need to say that this control is made in a completely autonomous way. Finally, the assistive AmI aims to assist the user in whatever he is doing. A good example would be an impaired or sick person, who has difficulties in going outside. An assistive system would check the refrigerator and order on-line the goods needed. Another example would be to provide the best route for a person in wheelchair, avoiding stairs or other obstacles.

The main advantage of this paradigm is that the user is the centre of AmI. For the first time, a computer system is built thinking on the user, with the objective of ensuring his well-being. The advantages are obvious. Computers always were powerful tools but when they become more than tools and start to “live” for us, the possibilities are boundless. The innovation in AmI is that the user doesn't need do go to the computer when he needs some service from it. In AmI, the computer is all around the user although disguised, and, in the ideal AmI, the user won't even have to ask what he needs. What the user gets from AmI is a better and simplified life. At home, in the office, in the hospital, in the mall, at the theatre, AmI can ensure the comfort and the safety of the people. At work, AmI provides tools, on-time information, support, the moment the worker needs it. The user gets an environment which shapes to his needs, constantly adapting, constantly searching for new ways to help him. More than that, the environment can be extended to the outside world, accompanying him to wherever he goes, providing services on the move using handheld devices.

These are advantages that users already take profit of nowadays. In the next section we will see what we can expect from AmI when it is fully developed.

2.8 The Vision

As Ambient Intelligence is a relatively new concept and depends on new technologies, it has yet much to grow. However, the potential of this paradigm is already widely recognized and the possibilities are appealing. The holy grail of Ambient Intelligence is to build an environment composed of Humans and "intelligent" artificial creatures, all working and living together in a very integrated way. These creatures will be living in the devices we carry around with us, in home appliances, in public places, in the clothes we wear and even inside our own body. They will all have large communication capacity, working so close to each other that they will appear like one global entity which is everywhere, taking care of our needs. Our interaction with these entities will be done in a way that is so natural to us that we will be interacting with them like we interact with each other, thanks to the developments in intelligent interfaces. We will be recognized in every building we enter and personalized services will be provided to us according to our preferences or needs.

This is the predicted and desired future of Ambient Intelligence. It will be reality when the technologies AmI depends on evolve enough so that their price lowers down and miniaturization and more efficient energy consumption policies allow for integration of more computational power into smaller devices. Many active investigation groups, supported by many interested companies are boosting the development of Ambient Intelligence, driving us closer to the accomplishment of the promises made by this amazing sub-field of Artificial Intelligence.

2.9 Ambient Intelligence Applied to Health Care

Ambient Intelligence has, as already seen in this chapter, many uses in a wide domain. However, when we think of the specific case of the health care, Ambient Intelligence can really make the difference on the users. The proof is the number of projects that keep appearing that try to merge Ambient Intelligence in the health care sector.

With its ability to assist people, we can think on the enormous advantages, mainly for impaired, convalescent, pregnant and elder people. In fact this sector and mainly the people involved have much to profit from this paradigm. With a constant monitoring from an Intelligent Environment it is possible to provide to this class of persons the most important: safety. The fact of knowing that despite being alone, if something happens to them help will come fast surely brings safety for the persons and their relatives. More than that, Ambient Intelligence can provide comfort. When interacting with the environment, these kinds of systems can change it so that it is always as close to the preferences or needs of the persons as possible. Being in an environment that constantly monitors them and responds in case of need, these persons can stay at their homes, with its routine and nevertheless live calmly.

But Ambient Intelligence in the health care sector does not stick just to home care. These kinds of systems can and have been successfully applied in environments like hospitals, leisure centres or elder houses. In these environments, they are not only enjoyed by the users but also by the persons working there, resulting in a better care being provided. We can think of a hospital environment where the vital signs of the patients are constantly being monitored and alarms raise in case of danger. We can also think of nurses scheduled tasks being dynamically adjusted according to the patient's state or location. Another example would be to automatically adjust the environment in the room of each patient according to its health status. But maybe, more important than all that is the fact of knowing at each time the context of the user. Having constantly knowledge of its location, the state of the environment around him and its own state may be very important. As we can see, the possibilities are infinite and result in many profits to all the persons involved. These profits should be deeply exploited so that we can make the best use of the technologies we have available nowadays.

2.10 Conclusion

Ambient Intelligence is a relatively new concept as the technologies it is based on are also recent. This may be the reason why there aren't many commercial implementations in use nowadays. In other hand, the growing number of AmI research projects proves that this is a very interesting field with a promising future. There are many different ways of looking at Ambient Intelligence. One of them and probably the more significant one from the point of view of this dissertation is AmI as the providing of services to its users. This is the vision supported by this work and presented in this section. We believe, as stated, that service based paradigms are the best way to implement dynamic and interoperable intelligent environments, as we will see ahead.

3 Ambient Assisted Living

Ambient Assisted Living can be considered as a sub field of Ambient Intelligence. It is more specifically dedicated to assist people on its day to day, mainly in the health care area. It builds on recent developments of IT and aims to promote the advantages that these technologies can have, mainly for the elderly. In this chapter, these technologies are detailed. By bringing together all these technologies, the result is a paradigm which has the best of each one working to assist the user [34].

3.1 Domotics

Domotics is a field which is dedicated to building automation and is also known as home automation. Its aim is to automate buildings, mainly in the more repetitive tasks that don't need our interaction. Domotics can be found in a wide range of domains: automatic watering and climate control in green houses, lights or climate control in big buildings like factories or malls, grass watering in a big city or in our homes. This is in fact the main use that this technology knows today: to automate our homes.

The objective when automating the home is to achieve first of all security and then comfort and energy saving. In the security field there are lots of applications, namely for access control, surveillance, intrusion, gas, fire or flood alarms, among others. In the comfort and energy saving field, the main task of the technology is to interact with HVAC systems as well as lights or window blinds to promote the comfort of the user and, if possible, to do it in an energy efficient way. Common tasks in this area are to maintain a specific level of temperature or luminosity inside determinate rooms according to the user preferences. Using domotics, one can set specific configurations or preferences and the system automatically takes care of interacting with the appliances in order to achieve that. More than that, the system tries to achieve the desired configuration in the most effective way, something that we humans sometimes fail to do.

Domotics by itself are not enough to build an Intelligent Environment since in Domotics there must always be a human to issue the orders to the system. The only change needed to put domotics to work with an Intelligent Environment is to replace the user issuing orders by an automatic intelligent system that acts on behalf the user and issues the orders to the devices on the house.

As for the technologies in domotics there are many standards being used depending on the type of environment pretended. In the field of home automation, the technology which is more used in Europe is X10. It uses the power line to carry commands to appliances and the possible commands are oriented to the home environment. The fact of using a network available in every house (the power line network) makes it easy to benefit of domotics and makes it easy to add new elements passive of being used by the system since it is only needed to plug them in the power line.

3.2 Pervasive Computing

Pervasive Computing is more than Ubiquitous Computing and refers to a recent trend in which computers are shrinking and hiding so much that in a near future it is expected that we interact with them without even seeing them or even without knowing it. The objective is to create environments that join in an integrated way humans and computers minimizing as much as possible the awareness of our interaction with them. Traditionally, computers have always been a tool which has its own environment, its own reality and humans have to transfer themselves to that reality and interact with them according to their reality. With this paradigm, the computer and human environment are to be seen as one.

The evolution started when computers began to be interconnected through networks, being the Internet the most visible example. At this point computers started to be ubiquitous but weren't however pervasive. When the mobile technologies developed, mainly on the areas of wireless communication and with the advances on miniaturization, Ubiquitous Computing became pervasive.

When talking about Pervasive Computing we must also talk about Mobile Computing and all the necessary support for interoperability and scalability, as well as about miniaturization and some embedded intelligence. We can therefore say that Pervasive Computing is made of four main areas which are: devices, network, middleware and applications.

Devices tend to be the common devices that we have nowadays, however with enhanced computational power and maybe smaller and with more functionalities. There are more and more devices that have wireless networking capabilities and environments have more and more devices available. Each day these devices have also more embedded intelligence which can be for instance the navigator assistant of our GPS which tells us the more economic path to our destination or the refrigerator which issues an alarm when its door is open for more than a minute. These are all small things that sometimes pass unnoticed because they are common but that start to build intelligent environments when all of them are interconnected.

Networks are nowadays very common and available and are needed for acting as a backbone for all these devices to be interconnected. However, as devices are growing in number and functionalities, networks must also grow. More than just grow, networks must also provide tools for achieving the interoperability that is necessary between these devices. This is even more important when we are nowadays talking of devices that can be so different on its functionalities or communication protocol.

The middleware layer is not much mentioned when one talks about Pervasive Computing however it is of great importance since it allows developers to build more powerful and faster applications. The objective of the middleware layer is to hide all the heterogeneity beneath so much different devices and show, instead of that, a single "programmable environment". One does not see the devices one by one, but rather sees the services that these devices can provide and works with them.

These applications are what gives life to the network of interconnected devices and allow for their functionalities to be fully exploited. Applications use the middleware layer and are usually very user centred which means that these pervasive environments will tend more and more to assist us.

In the future some major challenges are expected in this area. Maybe the more important one is how to deal with so many different devices and functionalities. Will our networks be enough? Other important evolution that is needed is in the miniaturization of batteries. In almost every mobile device, batteries are by far the biggest component which decreases significantly the possibilities of hiding these devices on the environment. We can for example think of wireless sensors which battery is far bigger than all the rest together, including the networking module. Another important challenge to address is how to deal with so much different architectures and types of devices, each one having its own protocols or interaction mechanisms. One cannot create high level APIs for interconnecting each new device with existing ones. In this field, maybe the best solution can be the adoption of standards, as long as there are only the needed standards.

Maybe one of the main contributions of Pervasive Computing to Ambient Assisted Living is the creation of enhanced environments that look just like normal environments. This is very important mainly when we are talking of environments that are directed to older people which sometimes are even technophobic, i.e., are very reluctant to accept new technologies and to use them. These pervasive environments can even be the only way of an elder person having such a computer system “controlling” its life.

3.3 Context Aware Computing

In Context Aware Computing (CAC), computers, as the name says, are aware of some context, i.e., they are able of reading and interpreting some context. So before continuing it is important to define what is context. Context can be defined from several points of view but ultimately it is accessory information that can be used to describe some object. In computing, context can be described by the uptime, bandwidth, network connectivity, available resources or users. If we think of ourselves, our context can be described by our location, the action we are doing, the people we are interacting with or even our social status. In our homes, the context can be described by factors like environmental parameters, appliances being used or noise level. The historic context is another one that is not much used by existing applications but that may be of importance: an object may be described by the changes that occurred on its context along the time and that information could even be used to predict the occurrence of events that tend to repeat with the time.

This information about context can then be used to develop what is called context-aware applications. These applications use the information about context to do its work. They can either react to changes in the context or they can provide the information to other applications that can make use of it. Then, how the information is used depends on the type and objective of the applications. If the application is dealing with an impaired person then one use could be to provide adapted services having in mind the impairment of the person. We could also think of applications that automatically adapt their interfaces according to the information about the social context of the person like the age or the present persons. If the context is the working environment of the person, then one use could be to assist the person on its job according to what the person is doing at each moment.

However, to acquire such information about context it is necessary to establish a bridge between implicit information that is out of the computer reality and transform it into information that can be understood and used by the computer. This bridge depends once again on the type of context but one important component that is usually present is the sensor. When we are dealing with a home context or even with the personal context, sensors are of great importance. They can provide information about environmental parameters inside the house and about the user health state like the heartbeat, the body temperature or the blood pressure. Information about the user’s social context can be retrieved from profiles that can store important information, even including user preferences. In the work environment, information about context can be retrieved for example from a glove which interprets the movements that the worker is doing [35] and forwards that information to the system, which then decides what to do. Information about context can even be retrieved from the user agenda or by scanning nearby Bluetooth devices for determining with whom the user is interacting or which persons are around so that specific services can be provided.

It is not difficult therefore to see that context is a very important concept in Ambient Assisted Living. Using information about context, services can be adapted to the person the environment is dealing with. More important than that, using information about the context of the user and its surrounding environment, the system has knowledge about its health and security.

3.4 Mobile Computing

The term Mobile Computing [36] describes computing devices that can be used while moving. This evidently opposes to what happened until now when computing devices resumed to computers standing still in our houses. This started to change in the 1990's, probably with the appearing of the first mobile phones and later the smart phones.

This field started with the need to move devices to better locations, i.e., with better resources for example. This movement can be either physical or logical: one can move a computer to a better location or change an instance of a user or an application to a better location. The difference is that a physical system can move anywhere while a logical one can only move to a different computer system. That computer system can however, be a mobile one. This of course raises lots of challenges to address. Entities must be identifiable while changing geographical location or while changing networks. And how to deliver a message to an entity that is constantly moving? When an entity is not in a network did it finish its execution or it just changed its logical location? These are the main subjects of investigation in this area nowadays [37, 38].

However, the trend that is more visible nowadays is that one can take computational power with us everywhere we go so that we can use it if needed, as we do in our homes or in our workplace. The most common examples are the notebooks and PDAs which have considerable computational power and portability. One can be in the train or in the car, travelling, and use these devices in a normal way, even connecting to a wireless network and having access to the Internet. The most recent advances in this field are in the area of Wearable Computers which intends to merge computers with our clothing so that they have embedded computational power. That is at the time, the most natural way of transporting computational power since it releases our hands from any device and enables us to perform our common tasks.

All the devices that are mobile depend however on a limited power supply. The batteries were the technology which enabled Mobile Computing and are nowadays the ones that are limiting its evolution. In fact, as mentioned in the previous section, batteries did not follow the pace of evolution of other technologies and nowadays still big and with not much capacity. This significantly diminishes the capacity of some mobile devices. Another important technology for the evolution of Mobile Computing was the communication technology. It evolved to a point where nowadays we can have connection to a network almost everywhere. It is common that public buildings have wireless networks that the devices can connect to and they are also common in our houses and workplaces. In the last case, one can connect to a HSDPA, GSM or UMTS network and be connected anywhere.

But what is the importance of Mobile Computing in the field of Ambient Assisted Living? The most important contribution is probably the wireless sensor. It would be impossible to constantly monitor the vital signs of a person if one did not have wireless sensors. With this technology, a person can be doing its day to day freely, while their vital signs are being monitored. Devices with this objective have even be already embedded in clothing so that all one has to do to monitor the vital signs is to wear an apparently normal jacket. Another important development in this field is the GPS sensor. One can have nowadays a GPS sensor embedded in a PDA which has also wireless networking capabilities and this way implement a wide range of services. These services are not merely monitoring the position of the person but also location based services [39]. These are services that change and adapt to the location of the person or services that become available or unavailable according to the same parameter. As an example, one can think of a city guide which is installed on a PDA and using the GPS signal points you to the closest interesting points, according to your preferences.

The evolution on this field will generate smaller devices with enhanced capabilities and portability which will enrich our experience with computer systems. Devices will get so smaller that we will wear them and even use them inside our body, radically changing the way we user computers.

3.5 eHealth

eHealth is the word that stands for electronic health which is the use of IT at the service of the health practices [40]. There is a wide range of services that IT can provide to the health care sector: Electronic Health Records (EHR), Telemedicine, Recommendation Systems, Group Decision Support Systems or mHealth [41], just to name a few.

EHR are records that store all medical information about patients so that, by following the same standard, that information can be accessed by different professionals in different institutions. By being electronic, this information can easily be shared and does not depend, as before, on the physical location of the patient file.

Telemedicine is the use of telecommunication technologies to approach doctors, patients and relatives. The main technologies used are telephone, video conference and the internet and the objective is, ultimately, to provide remote consulting or even remote examinations. In the simplest case, telemedicine could be a patient calling its doctor to solve a doubt and in the most complex case it would be a surgery locally executed by a robot which is remotely controlled by a doctor. Telemedicine is probably the most used field of eHealth, probably because of the technologies involved which are nowadays very common and available. In Telemedicine we can also talk about Teleradiology which is the ability to send radiographic images from one point to another. This means that a patient doesn't have to go to its doctor to take a radiography if it has a closer institution where he can do it.

Recommendation Systems (RS) are systems that are meant to assist the user in a determinate task it is doing. RSs are used in a wide range of domains but in this area, they are mainly used to, given a specific group of symptoms of a patient, retrieve a group of possible diagnoses. They are not supposed to take the place of the doctor but instead, to narrow the possible choices to a smaller number so that the doctor doesn't waste so much time choosing.

Recommendation Systems can also be used in Group Decision Support Systems (GDSS) [42] as more a member of the decision process, making suggestions according to the situation. A GDSS is a system that provides support in cases where a group of entities joins to take a decision. There will be different types of entities, each one with a different expertise area, and at the end a decision must be taken. However, the participants may not be in the same geographic location. They might even be in different time spaces so these systems are needed to support a wide range of cases.

Another recent trend that can be described as part of eHealth is mHealth: mobile health. mHealth is the name given to the use of mobile devices with the objective of providing health care to the patient on the move. With these technologies, the patient can be constantly monitored in real time, recurring to devices like PDAs and wearable sensors, which are connected to the network.

eHealth really made a revolution on the way that care is provided to the patients, making health care more accessible to all. In an AAL environment, the advantages brought by eHealth can be deeply exploited. Integrating these features into the environment brings great advantages, mainly when we consider an older or impaired population.

3.6 Conclusion

Ambient Assisted Living can be seen as a subfield of application of Ambient Intelligence, i.e., intelligent environments implemented with the objective of assisting users in their daily tasks, making the day-to-day easier. These assisted environments are based on technologies that allow them to be portable, pervasive and integrated in common environments. This means that they can be used in any environment, enriching it and empowering their inhabitants. This has numerous advantages for any kind of environment and especially for environments with people with impairments or limitations. In fact, the application of assisted environments with the objective of assisting these persons may mean the maintaining of their self-sufficiency and independence, increasing their well-being and confidence. We therefore believe that assisted environments are a strong answer to the challenges depicted in the first section, namely on how to provide care, support and security to the current ageing population while maintaining them at home.

4 An Architecture for AAL

When designing an Ambient Assisted Living environment there are some challenges that are raised by the heterogeneity and number of devices and technologies present. These devices must interact and exchange information and compatibility must be ensured between the different technologies and components. More than that, organization must be achieved so that an AAL environment is not just a bunch of devices interconnected but instead, a group of devices working together and sharing information towards a common goal. In this section we talk about the main technologies that can be used in a AAL environment and how to organize it from the architecture point of view.

4.1 Technologies Overview

The new technologies that are appearing allow for new ways of providing care to the elderly. They must be studied and their usability classified in terms of the well being, security and quality of life that each one or all of them together can provide.

4.1.1 Communication Technologies

Communication technologies are, as we have already seen, of major importance in AmI. They must be able of interconnecting very different components using very different infrastructures. They must provide security features since most of the times the information that is transmitted is classified. This is why, in this section, special attention is paid to technologies that help implementing the communication within the devices in an intelligent environment and with the exterior.

4.1.1.1 OSGi

The OSGi[7] Alliance [43] is a non-profit corporation created in March 1999 and is now formed by companies such as Mitsubishi Electric Corporation, Siemens AG, Oracle Corporation, Nokia Corporation or Sun Microsystems. Their aim is to promote interoperability of applications and services delivered and managed via networks. The OSGi Service Platform is a Java-based application server for networked devices, ranging from computers or mainframes to mobile phones or other handheld devices, which means it can be deployed to any platform running the Java Virtual Machine.

The use of this technology will let developers build java applications on a modular basis. The resulting modules are called bundles, which are not only competent to provide services, but also to use services provided from other bundles. In OSGi, a bundle can be installed, started, stopped or un-installed at run-time and without any kind of system reboot, which makes OSGi-based technologies very modular and dynamic.

The core component of the OSGi specification is the OSGi Framework. This framework provides a standardized environment for running computing programs and is divided into four layers:

- The execution environment, which is to be understood in terms of the specification of the java environment that is being used;
- The modules layer, which defines the policies for loading the java classes, having into account the inherent security issues;

[7] Open Services Gateway Initiative

- The life cycle layer, which adds dynamically the new modules and can start, stop or uninstall them;
- Finally, the service registry maintains a list of services as well as the bundles providing them. This layer is also responsible for dealing with the problems that arise from bundles being stopped or uninstalled while the services they provide are being used. If a bundle needs a service that is not currently unavailable, it is also the service registry that, when that service becomes available, will notify the bundle or bundles waiting on it (i.e., OSGi has the concept of event for more efficiently addressing these kind of problems).

This technology suits very well AmI since one complements the other. AmI can very simplistically be seen, as the providing of services to the user, thus there must be a way of providing them. OSGi comes up as a natural way of implementing AmI services, whether they are just java classes or something that runs hidden behind java classes.

The interest in such a technology is obvious since in AmI there are multiple entities communicating with each other, providing and requesting services, and there is generally also the need for controlling or monitoring the system from outside. All the advantages associated to Java (portability, modularity, etc) are fully exploited by the platform, granting it can be successfully deployed to devices with low resources and easily changed after implementation and during run-time.

4.1.1.2 HomePlug

HomePlug Powerline[8] Alliance is an organization composed of industry companies such as Intel, LG, Samsung, Motorola or Linksys that was created in March 2000. Their goal is to develop standards for promoting the use of the Power Line Communication: since most electronic devices we use are connected to power lines, it is possible to use this infrastructure that is completely implemented in our houses so that the devices can communicate. This system operates by impressing a modulated carrier signal on the power lines, with a frequency higher than the one used by the devices.

[8] Powerline is a technology that allows for data to be carried on regular power lines using a modulated carrier signal.

The Alliance works on three standards:

HomePlug 1.0 + AV (in-home connectivity, including digital home and consumer electronics applications, can transmit VoIP and HDTV);

HomePlug BPL (to-the-home, Broadband-over-Powerline applications);

HomePlug Home Automation (command-and-control applications).

Although all of those three aspects are interesting and useful, the one more useful in AmI is the third one as it is a low-speed, very low-cost technology, intended to control devices connected to the power line. Not only wired devices can be connected to this network as wireless ones can connect through wireless bridges that "spread" the signal over the air.

There are several advantages of adopting such technology. The most important is probably the fact of using an already implemented infrastructure since intelligent environments should pass unnoticed and be non-invasive. This would not only reduce the modifications in the original environment when implemented but also reduce implementation costs. Another great advantage is that in any house, the electrical sockets are in higher number than conventional network sockets, which means that there will be more points for devices to connect.

4.1.1.3 LonWorks

LonWorks is another solution for building a control network, this one developed by Echelon. This is an open technology that has been invented in the 90's and has been growing fast, mainly on Europe. It allows the transmission of data on a p2p basis over media such as twisted pair, power lines, fiber optics, and RF so we can have devices inside a home connected under different media, all of them using LonWorks protocol thus able to communicate.

Traditionally, a LonWorks network is built using *Neuron Chips.* A Neuron Chip is of main importance in this technology. Actually, it is a cheap (around US$ 3,00 a piece) chip, composed of three processors: two of them take care of the communication process and the other one is free for use as needed. This means that, when using *Neuron Chips*, we don't need to worry about implementing the communication in our devices and some processing can even be made in the chip. In case there is the need to make more complex processing, it can be connected to a more powerful processor through its IO channels, leaving the chip solely as a communication processor. This significantly reduces the implementation cost and complexity since all communication is automatically taken care of. This allows for example to connect a temperature sensor through a *Neuron* that constantly reads the temperature value and only alerts the system if the temperature is beyond pre-determined levels, thus reducing network traffic and system overload.

When adding, for example, a computer that has both a LonWorks connection and an Internet connection and the proper software for monitoring the system or realizing administrative work, we add the remote management feature required in intelligent environment. This is a technology that is very used nowadays in different environments ranging from cars to intelligent houses, from air-planes to traffic lights control, being especially useful in controlling environmental parameters such as temperature or luminosity, determining which devices are being used in each moment, if there are open doors, etc.

4.1.1.4 1-Wire

The 1-Wire network is a low-cost bus composed of a computer or microcontroller (the master) and one or more 1-Wire components (the slaves). These components are all connected to a twisted pair cable, using one wire to communicate and the other one as ground, which means it only supports communication in a direction at each time. Another important feature is that every 1-Wire component doesn't need an external power supply. It can drain the power it needs to work from the network (parasite power). This technology is a low-cost solution for connecting devices that do not need high bandwidth. The fact that every 1-Wire device has a unique address imprinted when it is created also adds an important feature that can be used in some cases (e.g. identification verification). Another important component in this network is the iButton. The iButton is a computer chip enclosed in a 16mm thick stainless steel can, which can be mounted in virtually everything since it is very resistant to adverse conditions such as scratches or humidity. It also has some memory so that information can be stored in it and, as well as any other 1-Wire component, a unique address. Data can be written or read to/from the iButton by just momentary pressing it against a Blue Dot (one of the iButton probes) which can be connected to a computer through a 1-Wire adapter. Adapters exist for USB, serial and parallel ports.

The iButton is nowadays mainly used as a key since each one has a unique number, together with personal information or security level access. In this specific case, it could be used not only as a key for opening the house door but also for storing, for example, clinical information about the person or to control home access or access level to a possible control interface of the home.

Concerning 1-Wire networks, they are wide spread nowadays and are used in a variety of fields. A large number of different sensors ready to connect to 1-Wire networks are available on the market or can be built. More complex devices exist, such an outside weather station, based on which recommendations could be made to the person leaving the house. With sensors that read humidity, temperature or luminosity, it is possible to watch environmental parameters inside the house and make decisions upon them. iButtons can also be placed on a window or door, allowing to know if it is open or close.

4.1.1.5 ZigBee

ZigBee is a low-cost wireless standard which allows the technology to be used in control and monitoring applications. It is also intended to be low-power which allows for a longer working time with smaller batteries. This protocol was created and is maintained by the ZigBee Alliance and its main uses are in the areas of home automation, hospital care and monitoring and control. The protocol is mainly intended for environments that require low power consumption and low data rates. It operates over radio frequency in the 2.4 GHz band and intends to be of lower cost than similar existing technologies, like Bluetooth.

Good examples of application are control networks which enable devices to be remotely controlled. This is useful in our homes for controlling our common devices as well as in any big building for controlling windows and HVAC systems. Another wide use that ZigBee has is in the field of wireless sensors. There are some implementations in greenhouses which have as objective to remotely monitor the environmental parameters inside the greenhouse [44] like temperature and humidity. The same standard can then be used to interact with the greenhouse windows and sprinkling system to maintain the ideal environment. The system can yet be used to monitor the path of perishable goods from the producer until the final consumer so that this knows that the good has not been exposed to harmful conditions.

There are three types of ZigBee devices: coordinators, routers and end devices. The ZigBee coordinators are the root of any ZigBee network and are also the most capable device. There can only be one coordinator per network. It is responsible for storing information about the network and for interconnecting with other ZigBee networks. The router, besides running an application, is responsible for passing information to other devices in the network so that data can go from one point to the other, through the several nodes. At last the most inexpensive device, the end device, can only talk to its parent device that can only be a coordinator or a router.

The ZigBee protocol automatically constructs the network as new ZigBee devices join. Typically it will create a cluster and bigger networks are organized as clusters of clusters. These clusters determine how the information will flow from one point to another with the objective of reaching the destination with optimized power consumption. Therefore, a node chooses the next node for transmitting the information according to the battery state and distance of the neighbouring nodes.

4.1.2 Ambient Intelligence

Ambient Intelligence (AmI) is a relatively new paradigm in information technology, in which people are empowered through a digital environment that is aware of their presence and context, being this environment sensitive, adaptive, and responsive to their actual needs, habits, gestures and emotions [45]. *AmI* can be defined as the merger of *ubiquitous computing* and *social user interfaces*. It builds on advanced networking technologies, formed by a broad range of mobile devices and other objects. Adding adaptive user-system interaction methods, based on new insights in the way people interact with computing devices (social user interfaces). As a result, digital environments can be created to improve quality of life of people by acting on their behalf. These context aware systems combine ubiquitous information, communication, and entertainment, with enhanced personalization, natural interaction and intelligence. The path to pursue, in order to achieve this goal, relies on a mix of different receptiveness from Artificial Intelligence, Psychology or Mathematical Logic (just to name a few), coupled with different computational paradigms and methodologies for problem solving, such as the conceptualization of figures such as the one of software agent, and its social counterpart in the form of Group Decision Support Systems [42].

4.1.3 Simulation

Simulation consists of creating an alternative reality to represent a real object. When making a simulation, generally one expects to predict how a given object or system behaves in the real world. This way, it is possible to draw conclusions about the behaviour of the system being studied and about its feasibility without having to actually build the system. It is very important to select the most important characteristics or behaviours of the system to simulate so that the results are as accurate as possible. These parameters to select may comprise not only the object being simulated but also the environment that surrounds the object. In the last years, this technique has grown a lot mainly thanks to advances in computer systems performances. Simulation is used to model complex systems that are either too expensive, too dangerous or simply impossible to assemble in the real world due to its characteristics. Some of the common uses of simulation are the modelling of natural systems (e.g. weather forecasts, storm evolution, earthquake damage), testing and optimizing new technologies, the construction of new or special buildings (e.g. skyscrapers, dams).

4.1.4 Group Decision Support Systems

In the last years we have assisted to a growing interest in combining the advances in information society - computing, telecommunications and presentation – in order to create Group Decision Support Systems (GDSS). Indeed, the new economy, along with increased competition in today's complex business environments, takes the companies to seek complementarities in order to increase their competitiveness and reduce risks. In this scenario, planning takes a major role in a company life. However, effective planning depends on the generation and analysis of ideas (innovative or not) and, for this reason, the idea generation and management processes become a crucial tool in present days.

The GDSS can however be applied to a new sector. We believe the use of GDSS in the Healthcare sector will allow professionals to achieve better results in the analysis of one's Electronically Health Record. This achievement is vital, regarding the explosion of knowledge and skills, together with the growing need to use limited resources more efficiently.

4.1.5 Ubiquitous Computing

Ubiquitous Computing (Ubicomp) is a new paradigm of human-computer interaction. Until now, this interaction has been very computer centred. A Human had to go to the computer and use it where the computer was placed, working with only one machine at a time, consciously of that interaction. Thanks to miniaturization and the development of the communication technologies, that is changing. In Ubicomp, the computation is spread everywhere, embedded in common devices. A user in an Ubicomp environment, ideally, would not even notice that he is interacting with a computer system. As defined by Mark Weiser "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it." [46].

In this paradigm, the devices constantly communicate with each other, mainly wirelessly. They can be portable devices and all of them have computational power. This way, we go from a paradigm in which the human interacts with only one device, consciously to one on which he interacts with a multitude of devices, without noticing it. These kinds of environments empower our living since they extend the capacities of the common devices or objects we use. These devices are tendentiously inexpensive, small networked processing devices. The common interfaces that have been used until now (e.g. command line, keyboard, GUI) are no longer suitable to be used in these new environments. There is therefore the need to evolve the interfaces and make them more user friendly and natural so that we don't notice that we are using a computer system. The evolution here are the intelligent interfaces, described in the next section.

4.1.6 Intelligent Interfaces

Intelligent Interfaces are the evolution of the common interfaces that now, besides making the human-computer bridge, provide additional features. The usefulness of these interfaces is bigger when the user doesn't exactly know how to really use the feature he wants or even when he doesn't know exactly what he wants. The request to the computer may be incomplete, incorrect, or not very specific and there must be ways to assist the user on reformulating the request or trying to guess what the user wanted. In order to achieve this behaviour, the interface should be adaptive, anticipate the needs of the user, be proactive and always explain its actions to the user. There are several examples of intelligent interfaces. One widely used today is the Intelligent Tutoring Service [47]. A tutor is a program used to provide education on a given topic. The intelligent tutor should permanently evaluate its student, adapting the contents, the speed of the lessons or the difficulty level to his needs and objectives. Similar to the tutoring service is the help service. This one however, instead of making the person learn something, intends to make the person do something. We can think also on search engines like Google or Yahoo as intelligent interfaces. In fact, they often do successful searches although we have misspelled the word or suggest similar words or similar concepts in order for us to improve our search. They must also filter the information, deciding which is closer to what we want and which is useless for us, in an attempt to meet our needs. Note that sometimes, even we don't know what or how to exactly make a search and the search engine must try to conduct us to more specific subjects to narrow our search.

The way these interfaces are implemented is also changing. Human-computer interaction first started with batch interfaces, evolving then to command line interfaces and later to graphical user interfaces. However, neither of those adapts to the new computer paradigms. As computation is starting to be less centred and more distributed (e.g. Ubicomp) interfaces will also have to change from being in the computer itself to being in the environment. Thus, the traditional keyboard and mouse will no longer be useful. The future of interfaces ranges technologies as speech recognition systems or gesture recognition (which includes facial expressions recognitions). These two are basically pattern matching used to map a voice or a gesture to a programmed action. We can also think of the new multi-touch screens and, more futuristic but already with some implementations, the direct neural interface, which is a direct communication pathway between a human brain and an external system. Any of these new ways of interacting with computers will be more natural to us, therefore more passive of being used by people which are not familiarized with computers or that have some impairment that prevents them to successfully use the common interfaces. Moreover, such technologies will increase our productivity and efficiency since we will be able to interact with computer systems without having to stand in from of them, and having our hands free for doing other things.

4.1.7 Multi-agent Systems

Multi-agent Systems (MAS) [48] emerged from the combination of Artificial Intelligence with distributed computational models, generating a new paradigm: distributed artificial intelligence. There are several valid definitions for a multi-agent system so we will try to define a multi-agent system from the AmI point of view. A MAS is a group of entities (may be software or hardware) which will "read" the environment they are in and make intelligent decisions in order to achieve some common goal (like the comfort or safety of people in the environment) based on knowledge from every agent in the system. For an agent to be considered so, it must show some basic abilities: autonomy, reactivity, pro-activity and sociability which means agents must operate on their own, read their environment and react accordingly, they must show initiative and take their own actions and be able to relate to other agents in order to achieve their goals. Additionally, an agent may show characteristics such as mobility, learning, veracity, emotions, among others.

In this paradigm, with simple agents or modules each one responsible for some part of the environment, the global objective is achieved through the cooperation of the agents [49]. They are able of taking their simple individual decisions completely independently and we see the result of these decisions as a global intelligent behaviour. This paradigm rapidly started to play a major role in the design of any intelligent system. A lot of research has also been done in the field of argumentation with agent technologies [42]. In argumentation, agents debate, defend their beliefs and try to convince the other agents into believing the same they do.

An example might be using a multi-agent system for controlling the environmental parameters in a house. Say we have an agent controlling the heat, another controlling the lights, another one controlling the air-conditioning system and other two software agents, one responsible for making the ambient comfortable and the other one for saving energy. When all persons leave some room the power saving agent votes to turn off the light and "everyone" agrees on that. However, the power saving agent also says that the heat should also be turned off because the room is empty. But, the comfort agent says it should remain on because someone could return to the room and will be happier if it is hot. Obviously, there is a contradiction: agents in the same system may have different objectives. Situations like this one are common in multi-agent systems. Not all the agents have the same objectives or, if they do, not all of them agree in the way they should be achieved. This is solved using negotiation [50]. In MAS, *negotiation* refers to the modelling of Human negotiation techniques so they can be used for conflict resolution between agents. In this case, they all could agree on turning the heat a little lower saving some power and, at the same time, keeping the room comfortable. This is just an example of the MAS potential and how fit the distributed architecture of this paradigm is to implement intelligent environments.

4.1.8 Computer Vision

Computer vision may be defined as the ability of computers to see their environment in a way that is close to the way we Humans see it. For us, vision is the sense where we get more information from and is, therefore, a very important feature to emulate in an environment that is supposed to interact with us in a natural way. Of course such an environment could react to our voice commands but it is even better if it is able to *see* us, see the objects around us, and perceive what we are doing, leading to actions that assist us without us noticing all that process.

Computer vision can be approached by some different technological areas such as mathematics, physics, neurobiology or robotics. We will here look at computer vision from the AmI point of view so it is expected that a computer vision system is able of recognizing home objects, people or movement so that it provides the system with very important data about the environment. This technology really started developing only a few years ago, when faster computers needed to deal with the amount of data generated started to be accessible. Tasks like object or face recognition for example are not so developed and it is difficult for a computer to identify some object in a picture. Detecting motion is easier for computer vision and is useful in an intelligent ambient for detecting if someone has broken into our house or in which division there are persons. Another interesting task with a wide range of applications that can be done using computer vision is perceiving emotional states by facial expressions. That however, is more in the intelligent interfaces field, although it uses computer vision. It is possible to estimate some parameters of a person's emotional state and program our system to react accordingly by, for instance, putting some specific music genre or delaying some reminder that is not very important. Additionally, computer vision can be useful for tasks as simple as showing on a screen the face of the people who is knocking the door or simply to watch our house from the internet.

4.1.9 Context Awareness, Machine learning and Reasoning.

Although these three aspects could be analyzed separately since about each one much can be said, they will be analyzed together as they are deeply related in AmI. In order for an intelligent environment to *reason* and choose what action to take, he must *learn* the characteristics or habits of the persons in that universe and learn from decisions he made in the past. For correctly learning about the person's habits, the system must be aware of the *context* (e.g. what surrounds the person, what the person is doing).

According to Dey [51], *context* is "any information that can be used to characterize the situation of entities". This concept has been around since ever and is rather difficult to define. It is however constantly used by us, most of the time, without noticing it. As an example, a conversation or a gesture sometimes is complemented by the context on which it is used. This way, we don't have to explain or detail everything since much information is contained in the context and doesn't need to mentioned, it is implicitly perceived. In computer science, the concept of *context awareness* means that a system has sensors that are able of reading the environment and thru that input, it understands what is happening and constructs its own representation of the environment. The concept of context is very important since only with an accurate representation of the environment the system will be able of making the right decisions. This representation is the source of all the information regarding the environment and the user. Using context, the systems can adapt the services they provide accordingly. For example, if there is a reminder scheduled of low importance and the user is asleep, the reminder will be delayed until the user wakes up.

Machine learning [52] is a subfield of artificial intelligence that aims to develop algorithms that allow computers to learn. Learning can be defined as the acquiring of new useful information. This happens with us Humans since we born until we die. Our brain constantly adapts and absorbs new information. In computer, learning has the same objective: to acquire new information that can later be used to improve the system. This learning can be done using a wide range of methods. Computers can learn by assimilating new cases in a Case-based reasoning (CBR) model [53], by modifying the weights in the nodes of a neural network[9] or thru evolutionary computation[10], just to name a few. CBR, by modelling our own way of learning from a high point of view and because it is possible to understand its decisions, has been a widely used method in many research projects involving learning.

Case-based Reasoning can be described as a problem solving methodology that relies on past experiences and knowledge to make present choices [54]. It is very common to observe that behaviour in us humans. One day, when we left the home and the sky was cloudy, the clouds turned into rain and we got wet. Nowadays, before leaving the house we look at the sky and, if it is cloudy, we take an umbrella with us. Basically, in this paradigm, one looks at the characteristics of past problems, the solutions adopted and the results observed and establishes an analogy with the current problem. If the problem is similar enough it is highly possible that applying the same solution will conduct to the same results.

Generally, a CBR process is divided into four different phases: Retrieve, Reuse, Revise and Retain [55]. On the first phase, the problem is analyzed and the cases that are relevant (i.e. similar enough) are retrieved from the memory. This measure of similarity depends on the problem domain but is generally a difference of sums of different values that characterize the case. In the Reuse phase, the solution from the previous case is mapped to the target problem, which may involve adapting the solution to some specific demand of the new problem. In the third phase, the solution is tested or simulated in an attempt to determine what happens. It is possible that the results are not as expected, which should conduct to the revision of the action taken. In the last phase, the solution adopted is stored in the memory, along with the new case that was experienced, contributing to the enrichment of the case memory.

Finally, *reasoning* [56] can be in a rudimentary way defined as a mental process by which one goes from a premise to a conclusion. Reasoning has been since ever a characteristic associated to Humans. Automatic reasoning is the name given to the implementation of

[9] A Neural Network is an artificial model of a biologic structure which intends to simulate the functioning of a biologic brain. It is based on interconnected nodes which values may change in order to change the behavior of the network.

[10] Evolutionary computation uses methods that simulate the biologic evolutionary mechanisms (e.g. reproduction, crossover) to solve problems.

reasoning methodologies in a computer. The objective is that a computer shows reasoning skills without Human interaction, in a completely or almost completely autonomous way.

4.1.10 Speech and Voice Recognition

Since it is the most usual communication and iteration method among Humans, the speech is of utmost importance for a system which intends to interact with us in a natural way. Speech recognition is essentially analyzing an electric signal acquired through a microphone and associating words to the electric signal. It is, therefore, classified as a pattern searching problem. In other hand, voice recognition deals with “who is speaking” instead of “what is being said”. When trying to identify the person who is talking, this technology uses the human’s anatomic characteristics (different throats or mouths produce different sounds) and behavioural characteristics (each person has its speaking style).

Both aspects are important and suitable for use in AmI. A system that accepts orders that are spoken by someone and supports this two features, is able not only to recognize the order but also to check if the person who gave it is authorized to perform such an order. For example, it is not expected that a visitor is authorized to change the heat in our house but he can ask the system about the weather outside or the time. This is even more important when it comes to persons with some special condition. If someone has movement issues, he can turn on the lights or the heat by talking, or call for help or even order the system to prepare a bath. This is all possible nowadays and is a very important component of an ambient expected to be intelligent and to interact with us in a natural way.

4.1.11 Wearable Computing

Wearable computing is a paradigm on which computational power is embedded into objects that can be worn on our body. The most common devices nowadays are wristwatches but new solutions are appearing rapidly. These wearable computers augment our reality and expand our possibilities, releasing our hands or our attention from repetitive tasks that can be performed by computers, allowing us to do more and better. Moreover, these devices usually work on their own so that we don't need to spare attention to them while they do their work. But wearable computing goes beyond helping in repetitive tasks. It can be a simple device travelling with us and acting as a personal assistant, providing us directions to some place, remembering us of some appointment. It can be a simple glove, equipped with sensors, that tries to guess what the user is doing based on finger and hand movement in order to provide real-time support. More complex examples are the recent military projects which aim to fully integrate communication, data processing, vital signs monitor, night vision and weapons system into a soldier's gear. This way, every soldier can be a node of a network, communicating with each other, knowing every time where the other soldiers are and constantly providing information to the base about his position, his medical state, etc.

However, in the ambit of this thesis, it is more interesting for us to study some examples of this technology in the medical area since there are lots of applications. ActiveECG [57] is an example of a cardiac monitor, with Bluetooth connectivity which can communicate with a computer or PDA in order to transmit data acquired from the person or transmit an alarm in case of emergency. It is able to monitor the cardiac activity and fire alarms when anomalous situations occur. Another good example is the Wealthy shirt [58]. Although it looks like a normal shirt, it is based on the concept of "intelligent clothing" and has enough electronics embedded on it for monitoring not only the heart beat but also other parameters like the body temperature or respiratory activity. In this case it uses GPRS to communicate with other devices and it can also fire alarms in the case of abnormal conditions.

4.2 Devices

Any Assisted Living Environment needs devices for the three main layers of its architecture: ubiquitous computing, ubiquitous communication and for interacting with the user and the environment. Each of these layers has different devices and objectives but only with a close integration of all of them the system can exist.

Starting from the bottom, with the ubiquitous computing layer. In this layer one can have a multitude of devices which have computing capabilities and that can work independently from each other. The existence of these kinds of devices is due to recent developments in the fields of miniaturization and communication technologies. It is nowadays common to have small and common devices which, externally, look exactly like they did a few years ago but now they have much more logic or processing power inside and even wireless networking capacity. One of the best examples is surely the mobile phone. Some years ago, a mobile phone was a big and heavy device which could only perform the task it was initially built for: make calls. However, with its evolution, the mobile phone got smaller and smaller and its functionalities grew while its size was reducing. Nowadays, our mobile phones are also calculators, agendas, enable us to play games and take pictures and videos and much more. The same happened with all the other devices like mobile computers, watches, home appliances and even window blinds have computing capabilities nowadays. What changed in our homes is that nowadays we have a large number of devices which have embedded computational power, even if we are already so used to it that we don't notice it. In Ambient Assisted Living Environments, there are many empowered devices that can be useful. One can think for example on the common computers, PDAs, displays, telephones, televisions, cars, photo machines, among others. Another very important category of devices are the ones used for monitoring vital signs which can raise alarms and communicate the values in real-time. These devices can be watches, wrist bands or even clothing. The need is therefore to take profit of all this power already present in our lives and put it to work in our favour. To do this, one needs to find ways to put all these devices to work together since "the whole is greater than the sum of the parts". This is where the communication technologies are important.

Communication technologies have been around ever since the first computers started to develop. There was always the idea that two computers sharing information could do more than when isolated. While in the beginning communication between computers was merely for academic research, this evolution ultimately lead to the today well known Internet. More recently, the innovation in this area lead to what today is known as ubiquitous computing. Ubiquitous computing has been developed based on the evolution of batteries, miniaturization and more energy efficient protocols which enabled to equip almost any equipment with wireless communication capabilities. Nowadays, the trend is to have devices which are constantly connected to a network, constantly on-line. One can have for example a PDA which at home, connects to our personal wireless network. When we leave the house, it automatically connects to a GSM or UMTS network until and if we go to a public mall which has a wireless network it can also connect to it. Devices have gained the ability to change between different networks with the objective of improving the networking experience and, at the same time, diminishing the battery consumption. In Ambient Assisted Living Environments, these recent evolutions can be very profitable. One has a very heterogeneous number of devices, which may have different mechanisms of communication. One can think for example in X10 devices which use the power line network working side by side with 1-Wire networks, Wi-Fi, Ethernet and Bluetooth networks, just to name a few. There is the need for all these different protocols to intercommunicate so that coexistence is possible and with that purpose there are many devices. One can think of the so common wireless routers that enable the communication between wireless and wired networks. There are also other types of routers which enable 1-Wire or X10 networks to directly interact with Ethernet networks so that commands can be sent from other networks directly. There is a whole infrastructure of communication devices that grants the necessary for very different devices to communicate.

Finally, the last layer of the architecture: the interaction layer. This is also a very important part of the system since the devices through which the user interacts with the system will create the image that the user will have of the whole system. This means that this interaction must be very user friendly. In fact, it aims to be as unnoticeable as possible: the ideal interface for such an environment would be one that one would use without noticing it. For achieving such a goal one relies in technologies that are as natural as possible for the user. These technologies include gesture, face and speech recognition. Some devices involved in this layer are therefore video cameras and microphones. Other common devices are touch-screens which are a recent development and allow a much more natural interaction than using the common mouse pointer. Following the path of the natural interface, a very important device in these kinds of systems is also the speaker through which the system can interact with the user through its natural language. Other devices for interacting not so much with the user but more with the environment are air conditioning devices, lights, window blinds, among others. It is through these devices that the results of the Assisted Living Environments are visible both to the user and in the environment itself.

4.3 Architecture

As seen in this section, the architecture for an Assisted Living Environment must, first of all, be able to accept a very heterogeneous group of devices and technologies. This means that the architecture must provide means for these components to coexist and work together. It is therefore mandatory that the architecture provides a communication and information sharing mechanism that all different components can take profit of as communication is a key component in cooperative systems. An architecture like this must also be highly expansible since new technologies and devices passive of being used appear every day. If it is expansible, it is also expected to be easy to expand, i.e. it should be easy to add new components to the architecture without having to mess with existing components or with the architecture itself. With new devices being added, the architecture must ensure the compatibility between all of them. It is also important that this architecture is scalable so that it can grow or even be included in higher level architectures like the ones already mentioned. At last, with all these features we can also say that the architecture cannot be static since much changes can occur so the architecture should also be described as dynamic. We can therefore enumerate the main characteristics of an Ambient Assisted Living Environment architecture:

- Dynamic;
- Modular;
- Expansible;
- Flexible;
- Scalable;
- Compatible;

These are the features that we will have in mind when developing this architecture. A technology that fits all these requirements is the OSGi technology. This is the technology that was chosen for implementing as the base for the architecture.

OSGi is an initiative that intends to establish standards in java programming, highly specific, catering for the sharing of java classes, that may be achieved in terms of a services platform paradigm [59, 60]. With OSGi we intend to build applications as an infrastructure to support a generic, platform independent framework. The result is a Service Oriented Architecture.

4.3.1 Adopting OSGi

Let us describe in this section the major role that OSGi will play in our architecture. There is a group of components, which are part of the home and should be connected to each other. This means that there must be a common language and a common communication channel so that all of them can share information and cooperate in order to fulfil their common goals. Our objective here is to connect a group of heterogeneous components in an integrated way and that is the OSGi contribution at this level.

When adapting the proposed architecture to fit the OSGi specifications we are faced up to some challenges, being the most obvious how to make the agents, used in some components of the architecture, OSGi compatible (since some of our components are agent based), and how to integrate in the same architecture components as different as sensors, databases and software. Besides that, agents can be very different between themselves, including the signatures of the methods they declare, so we must ensure that every agent is compatible with each other and with regular OSGi bundles. These issues and the solutions adopted are addressed in the following sections, where it is described how we have made our architecture OSGi compatible.

4.3.1.1 Multi-Agent Systems and OSGi

Adopting OSGi on each component of our architecture, forced us to find a way to make one`s agents compatible with OSGi bundles. The aim is to make accessible the functionalities of an agent (e.g., its methods) as services to other bundles. It would not be advisable to convert each agent into an OSGi bundle, since it would increase the development time and will throw away most of the advantages of Multi-Agent Systems based methodologies for problem solving. Hence, the verdict was to create an OSGi bundle that could make the bridge between regular bundles and Jade which is our agent platform: the MAS bundle. This bundle can deal with one Agent Container (AC) and implement the methods declared in the interface of the agents in that AC as its own services. Moreover, this bundle must be able to start and stop agents, which in practice, correspond to the start and stop of the services provided by them. The bundle, upon the reception of an invocation for an offered service from any other bundle, sends the invocation to the correspondent agent and delivers the respective result to the calling bundle. It must be noted that an agent, when trying to satisfy an invocation, may require the services provided by other bundles currently available. This is possible through the MAS bundle.

We are now ready to make a more detailed description of the MAS bundle. It has two methods for controlling the bundle that will be used, either by the client or administrator, in order to start or stop the respective bundle. Once this MAS bundle registers the services of the agents it creates, it declares the public methods of the agents on its interface, in order to make them visible to the other bundles as regular services. As for the interface between the MAS bundle and the Jade system, a JadeGateway agent (JGa) is being used. The task of this agent is to act as a bridge between Jade and non-Jade code. This agent is created when the MAS bundle is started, along with the other agents. The JGa has the knowledge of which services are provided by each agent running so whenever a request from a service arrives to the MAS bundle, it knows to which agent that request should be forwarded. When the request arrives, a shared object is created in the MAS bundle: the blackboard. This object has some fields such as the name of the service to be invoked or the content which is the response from the final agent. The MAS bundle simply fills the service name field, which is the name of the service that was invoked by another bundle. Then, the blackboard is passed to the JGa which, interacting with one or more agents, gets the answer needed to the invocation of the service. That answer is then written to the content field of the shared object and returned to the MAS bundle. The final part consists of the bundle resuming the invocation of the service by returning to the calling bundle, the content of the shared object. The bundle that requested the service will never notice all that went on while it waited for the result of the service invocation.

Likewise, if an agent needs to use a service from another bundle, it contacts the MAS bundle, which is responsible for contacting the correct bundle, invoking the service and forwarding the result back to the agent.

The more specific issue of the interaction between agents, inside each platform, is outside the scope of OSGi, and must be, therefore, addressed. Agent communication is indeed a very important subject since it implies directly with the performance and behaviour of the whole system. FIPA (Foundation for Intelligent Physical Agents) establishes several agent-related standards, being one of them the Agent Communication Language (FIPA-ACL) [61]. This standard defines how to syntactically and semantically construct a message which is the way agents share information. It is based on the concept of speech act.

The speech act theory states that through communication we can execute actions much like we do with our hands or our tools. As an example, if we ask a friend to close a window and he does so, we have closed the window just using our main communication mechanism: the voice. FIPA-ACL has a group of what is called performatives which represent a total of 22 speech acts. Each of these performatives represents a speech act that helps to describe a FIPA-ACL Message. Other parameters of a message are sender, content, conversation-ID, among other. The parameter *performative*, however, is much more important since it tells what the message is intended to, its purpose. Examples of performatives are Inform, Request, Not-Understood, Ask, among others.

The communication between the agents of this architecture complies with FIPA-ACL standard. By doing so, some drawbacks are solved and the compatibility of the architecture to foreign agents that follow the same standard is assured. At this point, any agent that complies with FIPA-ACL can run inside a container that is controlled by a MAS bundle, and may provide its methods as services.

The services provided by the MAS bundle are the following:

- AgentContainer startContainer() - This service starts a Jade container;
- void startAgent(name, agContainer, bundleContext) - This service starts a new Jade agent with name *name*, on the agent container *agContainer* and *bundleContext*;
- void stopAgent(name, agContainer, bundleContext) - This service stops the Jade agent with name *name*, on the agent container *agContainer* and *bundleContext*;
- BundleContext getBC() - This service returns the Bundle Context;
- void shutDown() - This service shuts down the Jade platform.

4.3.1.2 OSGi locally

Let us describe in this section the role that OSGi will play in the architecture. There is a group of components, which are part of the home and should be connected to the system. In one hand we have 1-Wire sensors, which give the system the knowledge about the temperature, humidity, luminosity and other factors. In another hand, we have the X10 network which allow for house equipments like lights or other electric devices to be controlled from a computer. We also have software components like basic decision mechanisms or monitoring modules and all of them have to like together in the architecture. Our objective here is to connect a group of heterogeneous components in an integrated way and that is the OSGi contribution at this level.

Inside the house, the sensors are connected to the central computer through the USB port and a bundle is responsible for constantly reading the values from the sensors and registering them. This bundle exports as a service the values of the sensors of the house, which can then be used by the rest of the bundles. There is also a bundle for each X10 equipment and each of this bundles exports as a service the commands that can be issued to the equipment it represents (such as ON, OFF, UP, DOWN, etc).

Let us assume that the air conditioning system has enough autonomy to control the temperature based on the client preferences. It would be a hard task for an X10 equipment to interact with a 1-Wire sensor and acquire information from it without intermediary equipment. With OSGi, the autonomy can be given to the bundle, which can easily issue X10 commands to the equipment based on the information that it obtains from the sensor bundle, without the need to know anything about the sensor, only having to requesting services. With OSGi, entities that are different may be easily integrated and put to work together ranging from sensors and actuators to house equipments, people or software agents.

Another problem we face when adopting OSGi is the compatibility between bundles. This problem arises from the very different bundles that we can have, aggravated by the fact that with OSGi, any bundle written by anyone can be incorporated into the system. As an example, imagine a class that exports a method which signature is a data structure declared within that same class. The bundle that made the call to the service will not “understand” the result. There is, therefore, the need to ensure the compatibility between all bundles. This was achieved by defining an ontology to be used by all the bundles. This ontology is a java package on which the classes relating to all the objects that can be used by the methods on its signatures are declared. If every bundle imports this ontology, the compatibility is assured and, if a bundle implements a new object is added, only the ontology must be updated. None of this would however be necessary if all the methods only used standard java classes (e.g. Integer, Char, String).

Having addressed these main issues, OSGi can be locally used in the components of our architecture simplifying its implementation. Moreover, OSGi provides a bundle that, off the shelf, allows for UPnP components to be viewed by the other bundles as services, extending its possibilities. Let’s look again at the home of the supported user: each bundle can provide local services like the control of the lights or the air conditioning system. If the user has, for example, an UPnP TV, its control will also be provided as a service to the user the moment he connects it. The fact of OSGi supporting UPnP devices has yet the advantages associated do the zero-configuration needed for including new devices, therefore, for including new services, which is especially useful for the elderly.

In the next section we will show how OSGi must be expanded so that it inter-connects the OSGi cells that are each component of the architecture, enabling the user to access remote services and maintaining the compatibility with the OSGi standard.

4.3.1.3 OSGi remotely

The fact that OSGi has a centralized architecture while this architecture is a distributed one, raises a last challenge that must be overcome. What we have is standard OSGi cells, each one being a component of the architecture. These cells, when interconnected, create a virtual organization, on a service-sharing basis. There must be, therefore, a way of accessing remote services of other cells since OSGi standard does not address this problem. R-OSGi is an extension to OSGi standard, which allows for a centralized OSGi application to be distributed, using proxy bundles. A proxy bundle is a bundle which provides not only a remote service exactly as if it was a local one, but also allows for local services to be remotely accessed.

The main idea behind R-OSGi is for remote services to be accessed by bundles like they would be if they were local ones, in a completely transparent way. What we do is to add an additional bundle to each cell that provides at last one remote service. This bundle is responsible not only for checking the service registry of the OSGi cell it is on, but also to search for services which should be provided remotely, announcing them on an external port. Remote bundles which want to subscribe its services will make a connection to that port and subscribe it. Moreover, each cell should also start a bundle for each service (or bundle of services) it wants to remotely access. This bundle subscribes the remote service as soon as it is needed and registers it on the local OSGi cell, as if it was the bundle providing it. Subsequently, any local bundle can use the service without the need to know if it is accessing a remote or a local one.

What happens is that, when a local bundle in cell A calls for some remote method, that call goes normally to the proxy bundle as if it was the one providing the service. After receiving the method call, the proxy bundle identifies the remote cell which is providing the service; let us call it cell B. It starts a connection with the proxy bundle of cell B and sends the method invocation. The proxy bundle in cell B receives and forwards it to the right local bundle (it may also need to connect to other remote bundles), which returns the result of the invocation to the proxy bundle. The proxy bundle in B then sends the result to the proxy server in A, which forwards it to the original bundle that called the method. The next figure is an example of a portable device, which connects to a remote server to satisfy a request of the user to plan a trip. The user, as well as the interface service on his device, does not need to distinguish if a remote service was used or not to get the results.

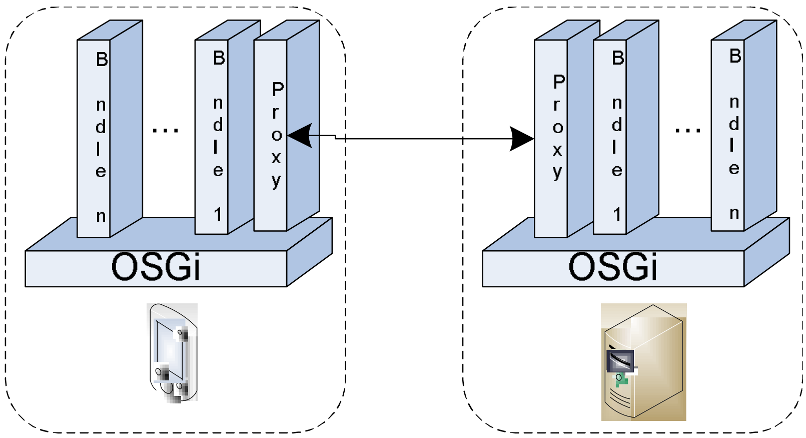

Fig. 4.1: Using remote services.

In Figure 4.2 we present an example of the architecture from the OSGi point of view. The OSGi network will be made of standard OSGi bundles (i.e., the OSGi cells), that may interact with one another. In this example, components in the area of the user body, such as a heart beat monitor, a handheld device or his personal assistant, are an OSGi cell, the devices and equipments of the user's house are another one, and the group decision along with the persons or equipment that make it, create even another OSGi cell. The red proxy bundles provide the local services of their cells as remote services, and the green proxy bundles provide a remote service as local services in their OSGi cells. In this implementation, the Personal Agent uses a remote service to plan the weekend, provided by the Group Decision bundle, when the user requests it. It also uses the lights service provided by the Home OSGi to turn the lights on or off, based on the location of the user. At last, the Temperature Monitor bundle uses the location service and the preferences of the user provided by the Location and Personal Agent to control the temperature at the home through the Air Conditioning bundle. Note that a service that is provided by an agent and is to be remotely accessed, is registered locally by the MAS bundle and remotely by the proxy bundle.

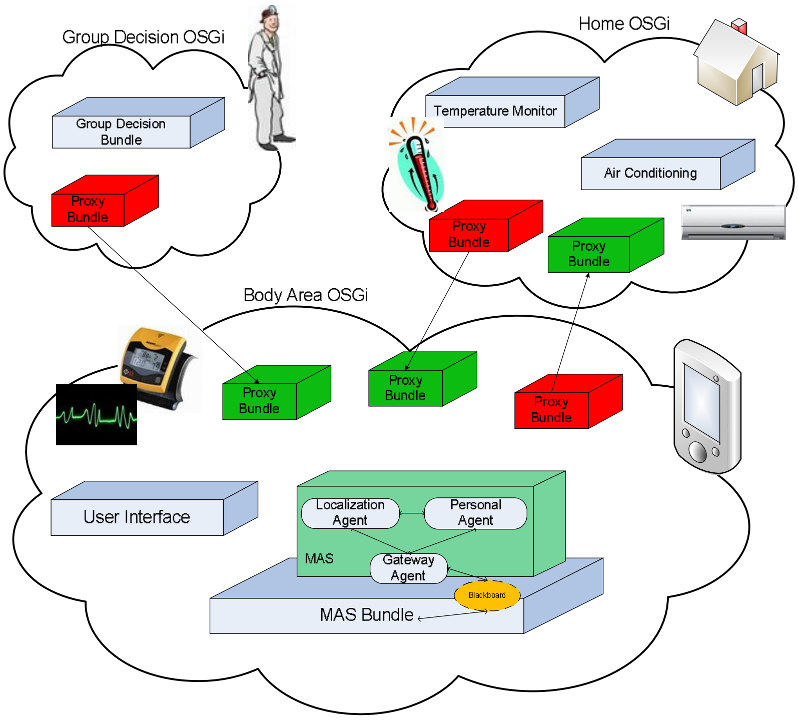

Fig. 4.2: An example architecture from the OSGi point of view.

4.3.1.4 Integrating Sensors

Sensors also have the need to an adaptation in order for them to be used in an OSGi architecture. These analogue sensors, as many others, do not directly provide the value they are intended to read. Instead, they provide a current value that must then be converted to the pretended value. This means that each sensor has its own way of working and one must know how to work with each one, how to convert the value read to Celsius degrees, lux, percentage, or others. This can evidently be a major obstacle to development teams, increasing the development time and costs since each developer must be an expert in possibly several sensors.

The solution can then rely on Service-Oriented Architectures, as is done in this work. With this objective in mind, some OSGi bundles with the functionality of interacting with sensors were created. These bundles provide the services of each sensor they are specialized in, making it easier to interact with the sensors.

Therefore, the temperature sensor bundle is able of communicating with the DS18S20 sensor through a DS9490R adapter, providing the following services:

- Float readTemperature(add) - This service reads the voltage value of the sensor with address *add*, converts it into Celsius degrees and returns the corresponding value;
- Boolean setResolution(res, add) - This service sets the resolution of the sensor *add* to *res*;
- Boolean setTriggerPoints(upper, lower, add) - This service sets the trigger points of sensor *add* to *upper* and *lower*.

Using these services, any external application can interact with temperature sensors without the need to know how the sensor really works, just invoking easy-to-use services.

As for the luminosity sensor, the bundle created interacts with the D2Photo sensor through a DS9490R adapter and provides the following services:

- Boolean readLuminosity(add) - This service reads the voltage of the luminosity sensor with address *add* and returns true if luminosity is high or false otherwise;
- Boolean reatStateChanged(add) - This service returns true if the state of the luminosity of sensor *add* changed since last reset or false otherwise;
- void resetSensor(add) - This service resets the state of the D2Photo.

4.3.1.4 Integrating the Database

The database, as the other components, had to be integrated into the architecture. This is because the database chosen does not provides services in the OSGi standard. The way that was solved was to develop an OSGi bundle that, using the jdbc library can connect to and interact with an Oracle database. When developing this bundle, another objective was in mind: security. One does not want to expose the database to external applications and allow them to alter the database in ways that can be harmful to the system. therefore, instead of developing a bundle that simply forwards sql statements, the decision was to develop a bundle that exposes as services the operations that can legally be performed on the database and that concern the sensors. Using this set of services, an external application can do every operation needed for managing the sensors on the database, nevertheless maintaining the security of the system. It has yet the advantage that developers do not need to know or use the sql language since they only need to invoke services and use the results returned.

Having these issues in mind, the database bundle provides the following services:

- void connect() - This service starts a connection to the database where the data regarding the system is stored;
- void addSensor(add, name) - This service adds the sensor with address *add* and name *name* into the database;
- void atachSensor(add, room) - This service adds information in the database stating that the sensor with address *add* is now placed in room room;
- void detachSensor(add) - This service removes the location of the sensor with address *add*. The sensor continues in the database but is not associated to a room;
- ArrayList getSensorsIn(room) - This service returns an ArrayList containing the addresses of all the sensors present in room *room*;
- boolean isSensorRegistered(add) - This service returns *true* if the sensor with address *add* is registered in the database, *false* otherwise.
- String getRoomFor(add) - This service returns the name of the room in which the sensor with address *add* is placed;
- String getTypeOfSensor(add) - This service returns the type of the sensor with address *add* is placed;
- ArrayList getAllSensors() - This service returns the addresses of all the sensors registered in the database;
- ArrayList getSensorsInUse() - This service returns the addresses of all the sensors that are currently in use;
- ArrayList getRooms() - This service returns the names of all the rooms of the house.

4.3.2 Compatibility between Bundles

Another problem we face when adopting OSGi is the compatibility between bundles. This problem arises from the very different bundles that we can have, aggravated by the fact that with

OSGi, any bundle written by anyone can be incorporated into the system. As an example, imagine a class that exports a method which signature is a data structure declared within that same method. The bundle that made the call to the service will not "understand" the result. There is, therefore, the need to ensure the compatibility between all bundles. This was achieved by defining an ontology to be used by all the bundles. This ontology is a java package on which the classes relating to all the objects that can be used by the methods on its signatures are declared. If every bundle imports this ontology, the compatibility is assured and, if a bundle is added which implements a new object, only the ontology must be updated. None of this would however be necessary if all the methods only used standard java classes (e.g. Integer, Char, String).

4.4 Communication

In the pursuit of compatibility, we cannot however forget that OSGi and R-OSGi are java based and we don't want to stick to the java language. We must therefore provide mechanisms of compatibility which are completely platform independent. To ensure the communication and compatibility with different components, the Web Services [62] paradigm was used. Web Services can be seen simplistically as a way of sharing information over a network and they are platform independent, being ideal for this kind of systems. Each component that provides information declares Web Services that are then requested by the other components which need to access that information. A component can, however, be at the same time a server and a client. A Recommendation System in this area, for example, uses Web Services provided by both the House and the Database and provides, as a service, the Recommendation that is used by a Group Decision.

The communication protocol is therefore of major importance. The information that is shared through Web Services is in XML format and what our Web Services share is FIPA-ACL messages represented in XML [63]. This FIPA standard allows a description of the main content of the message without having to read the content by using concepts like ontology, language or speech acts. This way, messages can be forwarded and sent to the final agents without the need to check the content.

However, we have defined a way of structuring the actual content of the message in XML. Examples of content are the temperatures in a room or in an entire house (a list of rooms), an aspect of the Electronic Health Record (EHR), the whole EHR or a recommendation coming from the Recommendation System. As an example of a sequence of communication, we can look at Figure 4.3. All the process is triggered by the bundle responsible for monitoring the vital signs of the Supported User. This bundle detects an irregular heart beat and warns the House central OSGi, where the MAS is running using R-OSGi. The MAS requests information from the movement sensors in another OSGi and asks again about the cardiac rhythm to the bundle that started the process to ensure that there was no reading error. Having gathered the information, the MAS decides that it cannot do anything to correct the situation and informs the Group Decision sending the anomalous values. This one contacts the Recommendation System which reads all the values of the sensors of the House and generates a recommendation which is then issued back to the Group Decision. After communicating with some more elements (like specialized doctors) and having in consideration the answer from the Recommendation System, two actions are taken: an ambulance is sent to the Home and the lights in the room of the user are turned on.

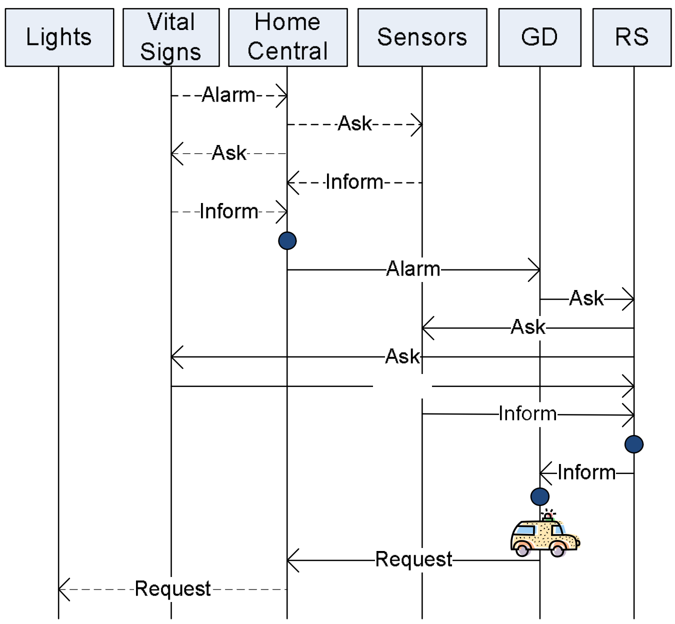

Fig. 4.3: Sample Communication Sequence Diagram.

In Figure 4.3 Dashed arrows represent R-OSGi services being invoked, regular arrows stand for FIPA ACL messages being exchanged through Web Services and circles represent some major processing or communication with local bundles using OSGi. An example of a XML FIPA-ACL message is depicted next. In this message, an external Group Decision requests through Web Service the values of temperature and movement in all the rooms. The service which provides the answer is the house central web service.


```
<?xml version="1.0"?>
<fipa-message>
 <act> request </act>
 <msg-param>
  <sender>
   <agent-identifier>
    <name> groupdecision </name>
    <addresses>
     <url>
    http://abc.com/groupdecisionwebservice
     </url>
    </addresses>
   </agent-identifier>
  </sender>
 </msg-param>
 <msg-param>
  <receiver>
   <agent-identifier>
    <name> house </name>
    <addresses>
     <url>
    http://def.com/housewebservice
     </url>
    </addresses>
   </agent-identifier>
  </receiver>
 </msg-param>
<msg-param>
  <content>
   <sensors room=”all”>
    temperature
   </sensors>
   <sensors room=”all”>
    movement
   </sensors>
  </content>
 </msg-param>
 <msg-param>
  <conversation-id>
   188273847728729
  </conversation-id>
 </msg-param>
</fipa-message>
```

4.5 Conclusion

Defining an architecture for an assisted living environment is an extensive work. There are many technologies that can be used, many paradigms and methodologies, a growing number of devices and many different ways of connecting them together. All these issues have to be addressed if we want to completely specify an architecture that can provide support to the services needed in an assisted living environment. In this section a large part of the development work was documented, ranging from the investigation of many different possibilities to the specific issues of implementing the technologies chosen to make them work together. Service based technologies have proven to be the more suited to implement this dynamic architecture and act as a bridge to connect all the different components and devices that will make up the assistance network

.

5 Simulation

Before implementing such a complex architecture it is advisable to create a simulation environment that allows for the system to be tested and assessed. In our case, we clearly need to study the behaviour of the system when specific cases occur, ranging from the reactive cases (e.g. react to a temperature change) to the more complexes (e.g. there is no movement in the last room the person was spotted for a long period of time). We also want to know what the house will do in the cases of malfunctioning of some components or if all the alarms went on at the same time. This is in fact one of the main advantages of simulation: it enables us to study specific scenarios that can hardly occur but are possibilities, without having to face the consequences of them really happening. We are therefore using simulation for studying the behaviour of the future house and improve it before all the equipment is acquired.

Our simulation consists of a fully configurable house, the environment around the house and the user [64]. When implementing this, the first step was to select which parameters to simulate. The main factors that exist in a real house and that influence the well being and the safety of its user are, in our opinion, important and need to be simulated. We are, therefore, simulating the temperature, luminosity, movement and humidity sensors, the fire, flood and gas alarm, the vital signs of the user which comprise the heart beat rate, the body temperature, the respiratory rate and the blood pressure and an the weather. In the actuators side, we are simulating all kinds of home appliances, ranging from an oven or a coffee machine to the lights or HiFi. It is also possible to simulate rooms with very different characteristics and even the actions of the user inside the house. The way these components are simulated is described in this chapter.

A major decision that we took was to develop all our simulation using OSGi and R-OSGi bundles (For detailed information about the simulation architecture consult the sections Simulation Architecture and The Reactive Bundle in Appendix A). This means that the current architecture and logic organization of the components is the same that the final system will have. By doing so, we expect to fasten the last phase of the development of the project since we will only need to replace the generation of the simulated data by the real components. The rest of the system remains the same and has already been hardly tested, which gives us great confidence on the performance of the final system.

5.1 Simulating the Environment

The environment is the main component of the system. It is where the user takes its actions, it is object of user interaction and directly influences the well being. We can think of our own homes or the places where we work. The characteristics of the rooms (e.g. insulation, percentage of glass, window blinds, capacity of air conditioning, etc) interfere with the environmental parameters inside the rooms. And of course, the environment inside the rooms directly interferes with our well being and safety. In a sunny place for example, the temperature inside a room will generally be higher if the percentage of glass in the walls of that room is higher or the insulation is poor. Even the geometry of the house can influence the environmental parameters: the temperature or humidity in a room influence the temperature and humidity in nearby rooms which have a door in common. The simulation of the environment is, therefore, mandatory and it is very important that it is fully configurable so that a wide range of different environments can be tested.

With that purpose in mind the first part of the simulation was created (Figure 5.1). The first step of the simulation setup is to set up the house configuration. A House is made of rooms. In the green area one can draw several rooms and dispose them in any way we like. When each room is draw, it receives a name and its characteristics must be configured according to four factors that we believe being of major importance: the level of insulation, the percentage of glass that covers the walls, the capacity of the air conditioning and the existence of window blinds.

These parameters where selected as being some of the most important in a house when one is worried about the environmental parameters inside the house. Insulation, glass percentage and air conditioning capacity are quantified from 0 to 100.

As an example, the temperature in a room with 0% of insulation would be the same as the outside temperature. In another hand, this implies that the temperature in a room with 100% insulation would depend only on the devices and user actions inside the house. This means that if no heat source is inside a room with 100% insulation, the temperature is the absolute zero. Of course, in real houses, 100% insulation is not a reality.

The air conditioning capacity represents, in a general way, the capacity of interfering with the temperature of that room. This might happen by the action of some air conditioning device as well any other heating or cooling device like central heating or gas heater. Meaning that with a 0% capacity of air conditioning, one could not influence the temperature inside the room while with 100% of air conditioning capacity, one could greatly influence this parameter.

The percentage of glass in a room represents the amount of glass that covers its walls. This influences the temperature and the luminosity inside the room. If a room has a low percentage of glass, it will be less influenced by the outside temperature, since glass usually has a higher heat capacity[11] than walls. In another hand, if the percentage of glass is higher, the temperature will be more dependent on the outside temperature. The luminosity inside the room will also be more or less close to the outside luminosity depending on the percentage of glass.

Another important factor that influences the luminosity inside a room is the existence or not of window blinds in the window, and that is the last configurable parameter in a room. One may say that a room has no blinds and the luminosity inside the room would depend only on the external luminosity and the lights inside the house. In another hand, if the room has window blinds, the luminosity would also depend on the position of the window blinds.

Additionally, one can establish connections between rooms that have already been draw. These connections represent the doors between rooms and have the same function in the simulation. They are used in two cases: to simulate the movement of the user inside the house and to simulate the influence of environmental parameters that each room has on the nearby rooms.

[11] Heat capacity is a measure of the energy required for change the temperature of an object by a certain interval.

At last, one can save or load a previously saved design so that each time a new simulation is started one does not have to draw a new house configuration.

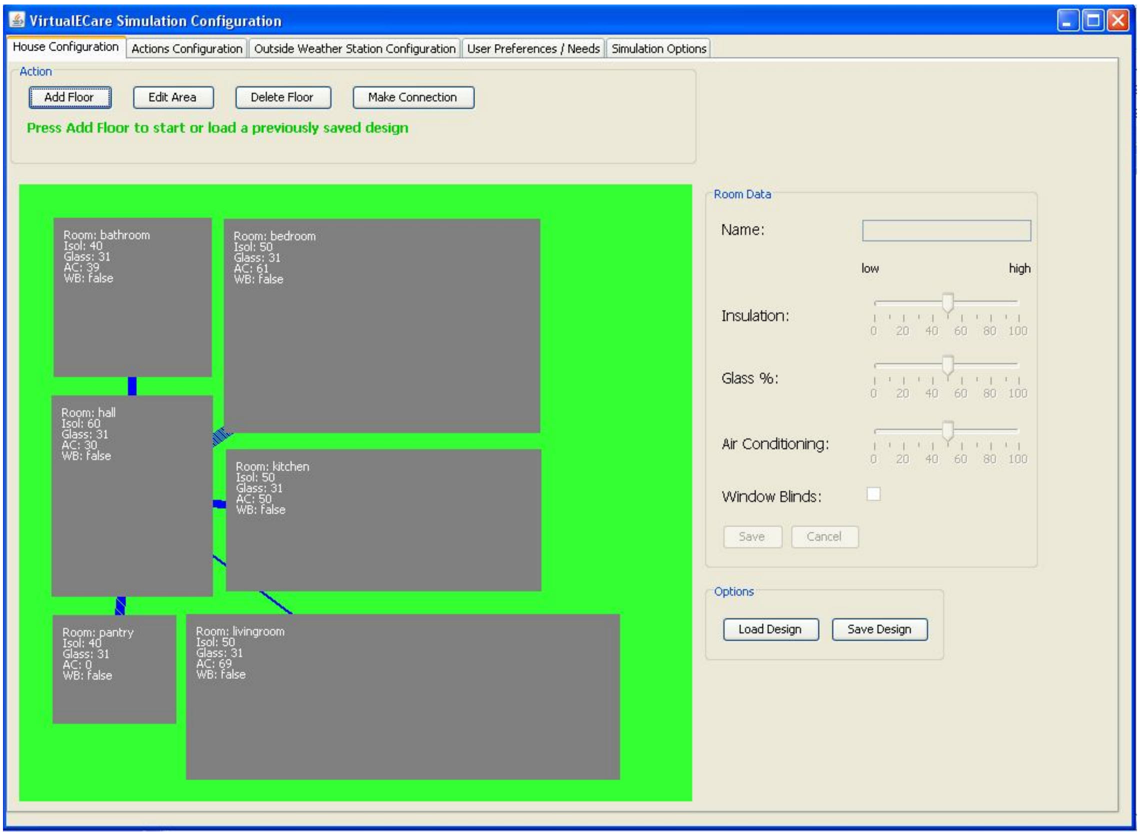

Fig. 5.1: The Simulation Environment Configuration. Some rooms are draw and some connections between them are visible in blue.

5.2 Simulating Sensors

For generating the data concerning the environmental parameters, there is a group of different sensors that are being simulated. There is an outside virtual weather station that will be later replaced by a real 1-Wire pre-assembled one. This weather station (as the real one) can provide information about the wind speed and direction, temperature, humidity, barometric pressure, rain fall, sunlight intensity and lightening. With this we have access to the knowledge regarding the environment around the house. In the real case, this is useful not only for informing the user but also to be accessed by Group Decision Support Systems, when deciding some recommendation for the client. In fact, all the simulated data can be remotely accessed by external entities, like the VirtualECare GroupDecision module. In the simulation, this weather station has yet another very important role: it is the base of all the simulation. This means that the simulated parameters inside the house depend, in first hand, on the values of the same parameters outside, much like what happens in the real life. The exact way that the external environment affects the inside environment depends then on factors like insulation, house exposure to sunlight, among others.

The temperature and luminosity sensors are distributed along the house to monitor these parameters. They will be replaced by the DS18B20 and D2Photo 1-Wire sensors. The humidity sensors are used mainly in the bathroom, to detect that the user is having a shower and in the rest of the house, to monitor the air humidity. They will be replaced by the TAI8540A 1-Wire Humidity Module. The fire, flood and gas sensors as well as the movement sensor, in other hand, are X10 based. The fire, flood and gas sensors are used to detect dangerous situations that threat the life of the user and the real ones will, respectively be, the PR8307, PR8306 and PR8808. The movement sensor is used to determine in which room the user is in and the real one will be the PR8070.

As for the vital signs of the user, we intend to use the Vital Jacket developed in the University of Aveiro. By now, we are simulating the values of the heart beat, the body temperature, the blood pressure and the respiratory rate. This data is mainly to be used remotely by doctors or other services and to raise alarms in case of dangerous or abnormal readings.

So the question here is now how to generate the values. Should they be random? Should they follow a function? Should it be a mixture of both? We cannot forget that we also want to create specific scenarios. So, we need to find ways to both set the values manually and to let them evolve in a natural way.

As said before, the base of the simulated sensors is the outside weather station which means that the values inside depend on the values outside. The only configurable sensors on this simulation are, therefore, the weather station sensors. The way we choose to implement scenarios uses XML files. Each XML file starts with the address of the simulated sensor and is followed by a list of pairs of the type <tick, value>. Whenever a sensor bundle is started in the scenario mode, it searches for XML files that refer to any of the virtual sensors it controls. Let's say that we are talking about a temperature bundle and it finds a XML file that refers to temperature sensors it controls. The bundle starts counting the seconds since it started and whenever the value of the seconds passed is equal to tick, the value of the simulated sensor becomes value. As an example:

```
<?xml version="1.0"?>
<scenario>
 <address>
      1AB2CD44200A1
 </address>
 <events>
      <value tick = "500">
           20
      </value>
      <value tick = "1000">
           22
      </value>
      <value tick = "1500">
           24.5
      </value>
 </events>
</scenario>
```

This XML file would set the temperature of the sensor to 20 degrees 500 ticks after starting the bundle, to 22 degrees after 1000 and to 24.5 degrees 1500 ticks after starting the bundle, remaining like that until the bundle is stopped.

The same methodology is used for any of the other sensors being the only difference the values (that must be according to the type of sensor). Now the second case, when we don't want to set up static scenarios. Instead, we want to let the values flow and see what happens. In this mode, the values are generated using a Gaussian distribution. This distribution was selected due to its ability to shape natural phenomena. So, when configuring the simulation, in the "Outside Weather Station Configuration" tab, one selects for each simulated parameter the mean value and the variation that may occur and that will define the weather that will occur during the simulation (Figure 5.2). It is therefore possible to create a more stable weather or, in another hand, to create a weather configuration that can change rapidly and unexpectedly.

Let us see now how the values inside the house are generated. The first step is to identify the main factors that stand between the outside environment and the inside environment in our homes. These factors were already mentioned on the previous section and are the factors used when configuring each room (e.g. insulation, glass percentage, etc) so the values generated depend on these factors. There are two additional factors that were used when generating some of the parameters: the previous values and the number of adjacent rooms. We realized, when developing the simulation tool, that the temperature inside the house would change immediately after the state of the air conditioning changed, which does not corresponds to the reality. In our homes, when we change the state of the air conditioning, the environment takes a while to change and the factor time is used to add some delay to parameters like temperature or humidity. Luminosity, in another hand, changes instantly so the factor time is not necessary. As for the number of adjacent rooms, it is known that the environment in a room is affected by the environment in the rooms that share a common door and, in another hand, it also affects the environment in all those rooms so this is another factor to consider. As an example, one can look now to the mathematical formulae for generating the temperature inside a given room:

```
    if (ACservice.getAirCondicioningConfig(room).isOn())
    temp = (((100-airCondCapacity)/100)*0.5*prevTemp +
(airCondCapacity/100)*airCondTemp +
    ((100-airCondCapacity)/100)*0.3*actionTemp +
    ((100-airCondCapacity)/100)*0.2*consTemp)-
    ((100-insolation)/100)*(((100-airCondCapacity)/100)
*0.5*prevTemp + (airCondCapacity/100)*airCondTemp +
    ((100-airCondCapacity)/100)*0.3*actionTemp +
    ((100-airCondCapacity)/100)*0.2*consTemp);
    else
       temp = (0.5*prevTemp + 0.3*actionTemp +
0.2*consTemp) –
      ((100-insolation)/100)*(0.5*prevTemp + 0.3*actionTemp
+   0.2*consTemp);
```

In this formula:

prevTemp – The previous value of the temperature in the room;

airCondCapacity – The capacity of the air conditioning provided as a parameter on the environment configuration;

insulation – The insulation of the room provided as a parameter on the environment configuration;

airCondTemp – The temperature of the air conditioning on the room provided as a service by the actuators bundle.

actionTemp – The temperature resulting from the action that the user is currently executing;

consTemp – The contribution of the connecting rooms to the temperature of this room.

The simulation of the effects of the outside weather in different types of room is achieved by assigning different characteristics to the rooms.

As another example, we can look at the luminosity. In this simulation, the luminosity depends on three factors: the outside luminosity, the position of the window blinds and the fact of the lights being on or off. The values of the outside luminosity, in its normal mode, are also generated by a Gaussian function and are provided by the weather station bundle. The state of the lights and the window blinds is a service provided by the X10 bundle. So, if the lights are on, the luminosity level is always high. Otherwise, it depends on the existence of windows and, if they exist, the position of the window blinds. If they are down or no windows exist, the luminosity is low. The luminosity function is as follows:

```
    if (hasWB)
    {
     if (LightsService.isLightOn(room))
        lum =(WBPos/100)*WeatherService.getLum(tick)+
lightsIntensity;
     else
        lum = (WBPos/100)*WeatherService.getLum(tick);
    }
    else
    {
     if (LightsService.isLightOn(room))
        lum = WeatherService.getLum(tick)+lightsIntensity;
     else
        lum = WeatherService.getLum(tick);
    }
```

where wbposition is the percentage of the position of the window blind (e.g. a window blind that is at the middle of the window corresponds to wbposition = 0.5). Similar processes occur for the rest of the sensors. While the weather station is ruled by Gaussian functions that can be configured when the simulation starts, the values of sensors inside the house depend on those external values and the other factors mentioned.

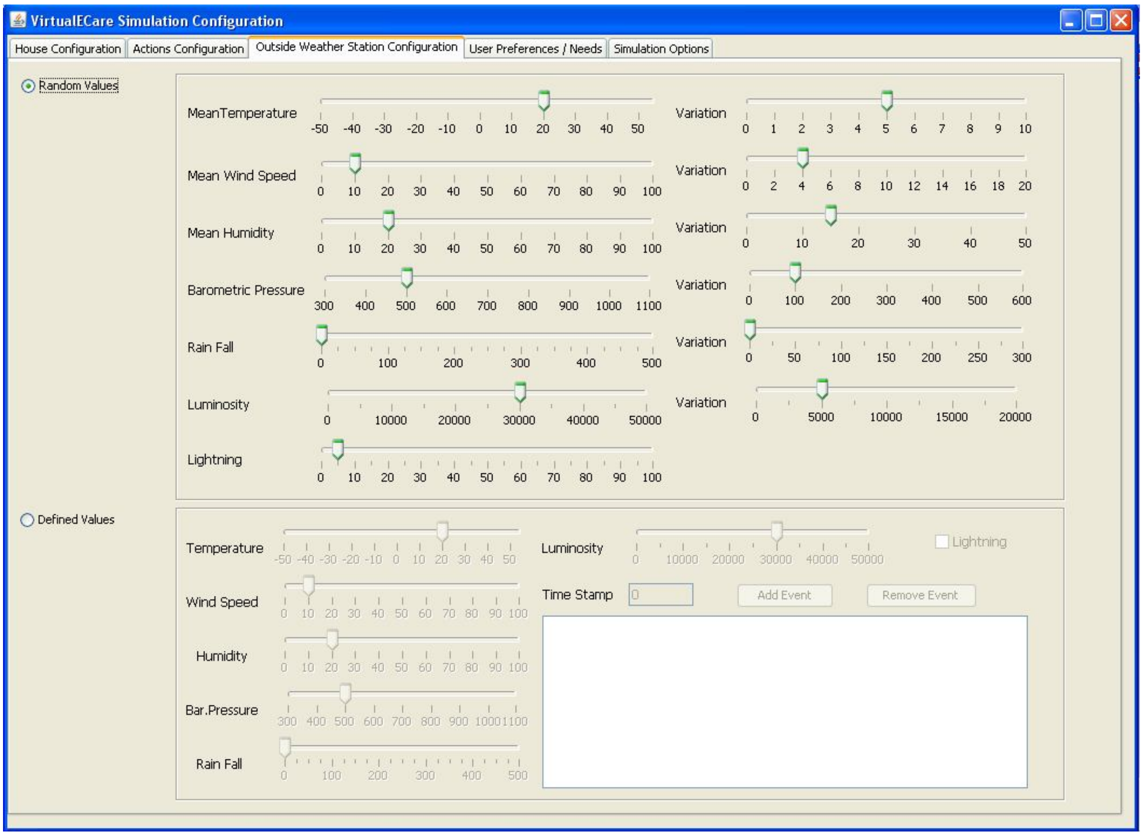

Fig. 5.2: Configuration of the outside weather station.

5.3 Simulating Actuators

The actuators in the house will be responsible for controlling the appliances. As said before, they will be X10 actuators. However, the actuators must be simulated in order for the actions resulting from the simulated sensors to have effect. We are therefore simulating the lights, the window blinds, coffee machines, HiFi, televisions and the air conditioning systems. The simulated lights enable us to send commands to turn the lights on or off. The window blinds simulation enables us to pull the blinds up or down. The control of the coffee machine, HiFi and television are essentially used to turn them on or off according to the policies of the house (e.g. when the room is empty, at given hours). As for the air conditioning simulation, it allows us to turn it on, of, more cold and more hot. In our simulation, all of these virtual X10 actuators allow for their state to be checked i.e., we can read the position of the window blind or the actual temperature of the air conditioning system. Although this is possible in some recent X10 modules, it is not possible on the older ones.

As we did with the sensors, we defined two ways of simulating the virtual appliances. We can either generate specific scenarios or they can run in what we call normal mode. In the scenarios, the methodology used is the same that was used for the sensors. We create a XML file for a specific X10 address which represents an appliance and that has a list of valid states. In the correct time, the state for that appliance is active. In the normal mode, the actuators behave much like they behave when controlled by a human. This means that their state will only change when told so by the system, unless there is some malfunction. Therefore, when the system tells the lights to be on, they will remain on until told otherwise.

5.4 Simulating Users

In every environment, there must be users for all this to make sense. More than that, the users interact with the system and are, probably, the most unpredictable part of it. A user can change the environmental parameters of the house he is in by using the actuators or through the common action he performs on the house. As an example, if the user decides to take a bath he is increasing the temperature and humidity on the bathroom. The simple fact of interacting with certain devices interferes with the environmental parameters: if the user turns on the oven to cook a meal, the temperature in the kitchen will rise. This justifies the importance of simulating the user and its actions inside the house.

In the simulation tool that was created, it is possible to simulate a wide range of common actions that one generally performs inside a house (Figure 5.3). There are three running modes for the user actions simulation: Full Random, Bounded Random and Planned.

In the Full Random mode, one has no control over the user actions being performed inside the house. The only restriction is the user level of activity which dictates the rate at which new actions are generated when no action is being performed and the maximum length of the action which is 20 ticks. The resulting behaviour is actions being performed in random rooms, with random lengths and starting at random ticks. In this mode, the oddest things can occur, like the user having a bath in the hall.

In the Bounded Random mode, the actions are generated randomly, with random lengths, just like in the Full Random mode. However, in this mode, it is possible to configure which actions can be performed in which room. This mode enables a more realistic simulation without however having to worry about completely specifying what is going to happen in what tick in which room. When configuring the simulation, we choose this mode and say that in the bathroom the user can only take a bath and make exercise, then only these actions will be generated for this room, having however random lengths.

The last mode is the one where we can completely specify what we want to happen in the simulation, concerning user actions. In the Planned mode, one can select which action will be performed in which room at each time with what length. This way, it is possible to completely specify the actions of the user during the simulation. As an example we can look at Figure 5.3, where Planned Mode is selected. When the simulation starts the user will be cooking in the kitchen for 40 ticks. After finishing, the user will be in the hall doing nothing during 10 ticks. He will then be watching TV in the living room during 100 ticks and exercising in the same room for 80 ticks. At last, the user will be taking a bath during 30 ticks in the bathroom.

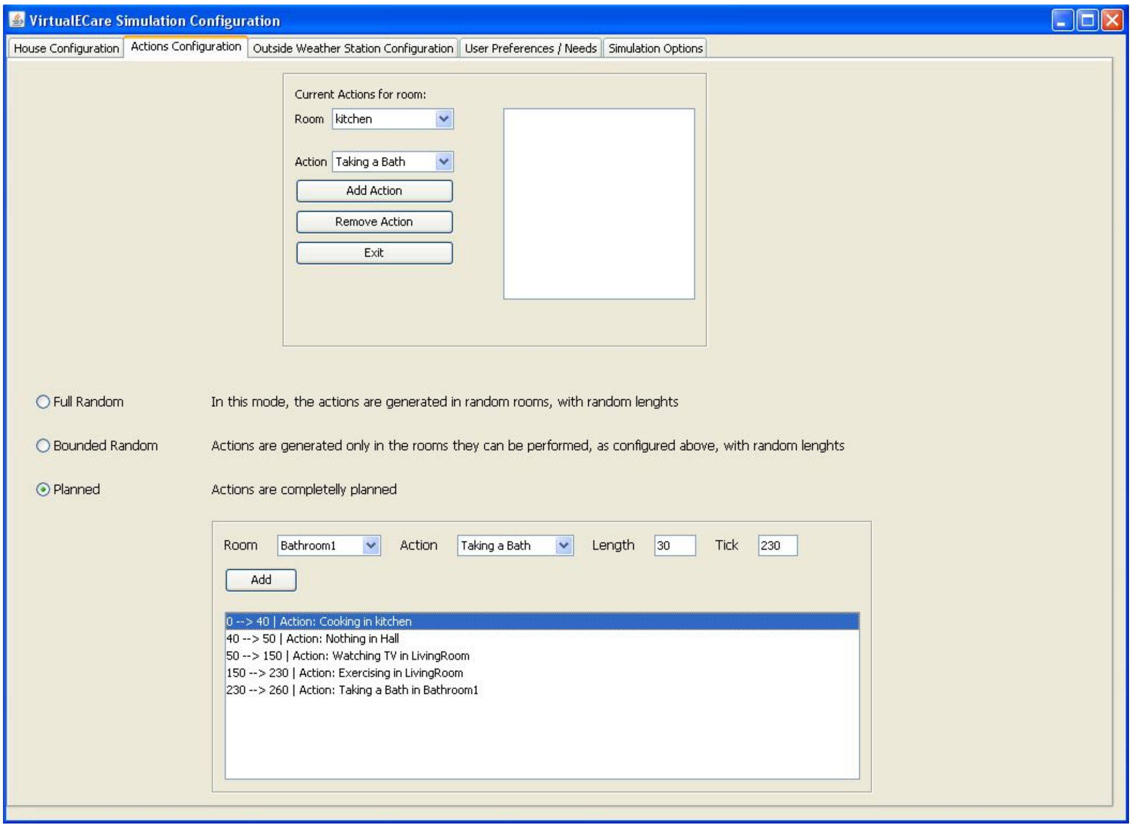

Fig. 5.3: User Actions Configuration: Planned Mode is selected.

There is, however, more important data to be simulated concerning the user. As the final system aims to monitor the user vital signs these should also be simulated to test the inference mechanisms that try to evaluate the health status of the user. By simulating the vital signs, one can cause specific cases to occur and see how and how fast the system reacts to certain vital signs configurations and this way improving these inference mechanisms. When configuring the User Vital Signs two modes are possible: the Random mode and the Planned mode. (Figure 5.4)

In the first mode, the user vital signs can be configured to run randomly, inside pre-determined values. When using this mode, one selects the mean value of each vital sign and the variance it may have. The values are then generated using a Gaussian distribution that uses these values as its mean and standard deviation. The decision on using this probability function was due to its performance when used to simulate natural phenomena. As an example, if we configure the heart beat mean to 80 bpm and its variance to 20 bpm, then we would have a simulated heart beat between 60 and 100 bpm, being however a value around 80 much more probable to happen than one around 60 or 100.

In the Planned Mode, the vital signs are completely planed and it is possible to determine the exact value of the vital signs at each moment. With this mode we can for example simulate a heart attack for a given time and see how the system reacts to it.

The final values of the vital signs are however not the ones that are generated here. The simulation takes these values and modifies them according to the action the user is performing at that time. This means that if we configure the simulation of the heartbeat to a mean value of 80 bpm and the user is exercising, the heartbeat observed in the simulation will be considerably higher during the time the user is exercising. In another hand, if the user is sleeping, it will be slightly lower.

It is important that the system knows which actions influence which vital signs. By knowing that a certain activity influences some vital sign by a certain amount, we can advise the user to stop doing it or even to take another action that leads to a better state of the user. As an example, in the simulation, the action *exercise* increases the heartbeat, the body temperature, the respiratory rate and the room temperature and humidity. Therefore, if the heartbeat is too high and the user is exercising, the system can advise it to stop or even go rest so that its heartbeat can lower. Similar information is associated to each action that is simulated.

Additionally, some more information about the user can be provided. There is the user activity level, already mentioned before, that determines the rate at which new actions are generated when no action is being performed. The user level of richness is used by the system as a factor on the decision making process. When deciding which action to take when the temperature rises in the exterior, the system can decide to lower the window blind or turn on the air conditioning. The most effective action would be to turn on the air conditioning. However, if the user level of richness is low, the system could opt for acting on the blinds. It is at this level that this parameter is used. As for the user needs and preferences, the use is much the same. The user might like a colder environment but the doctor also has a word to say and it might be better for him to be in a hotter environment. When the system acts on the environment, all these factors are weighted to try to find the optimal solution that provides both the comfort for the user and the adequate environment. The third factor is the economical one that is, as said before, contained in the user level of richness parameter. This basic decision mechanism was improved as will be seen ahead.

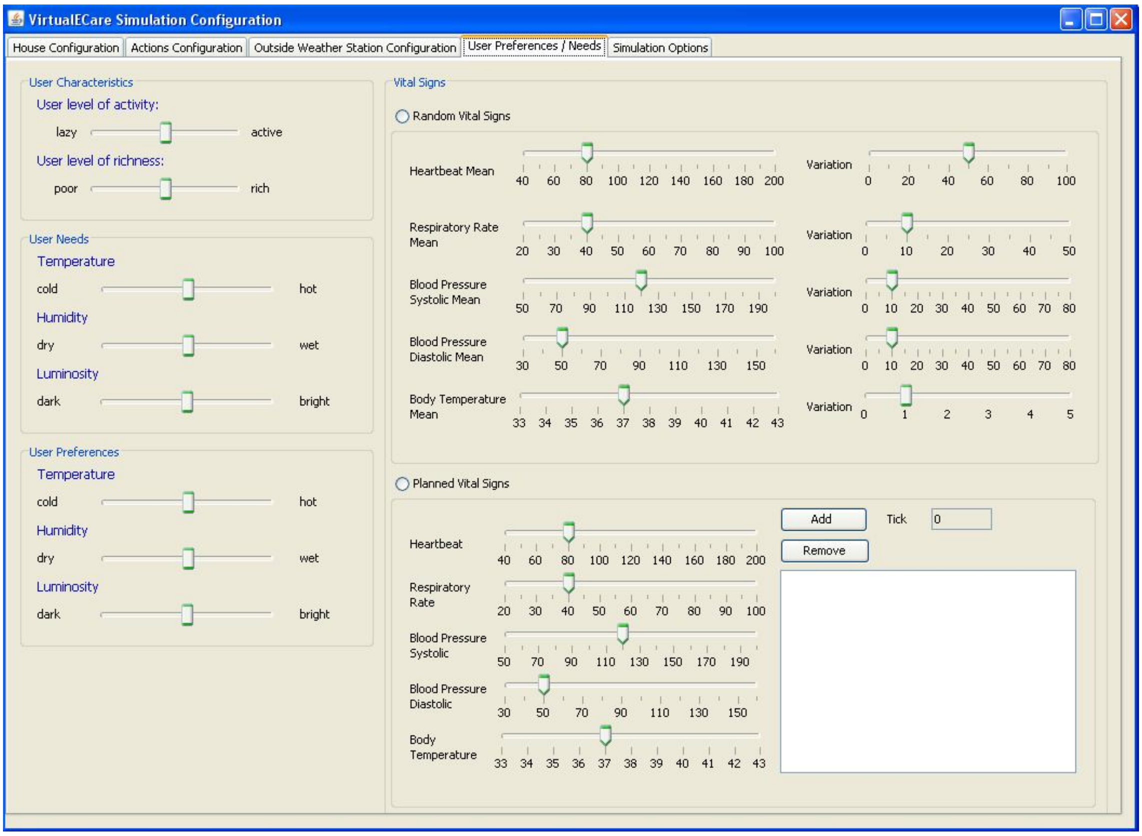

Fig. 5.4: User Parameters Configuration.

5.5 Running the Simulation

The last phase in the process of configuring the simulation is the Simulation Options tab (Figure 5.5). In this tab, one can set the simulation speed as well as its length. A log file can be saved according to the desired verbosity. The values at which alarms fire can also be set here. Finally, when everything is configured, the simulation can start.

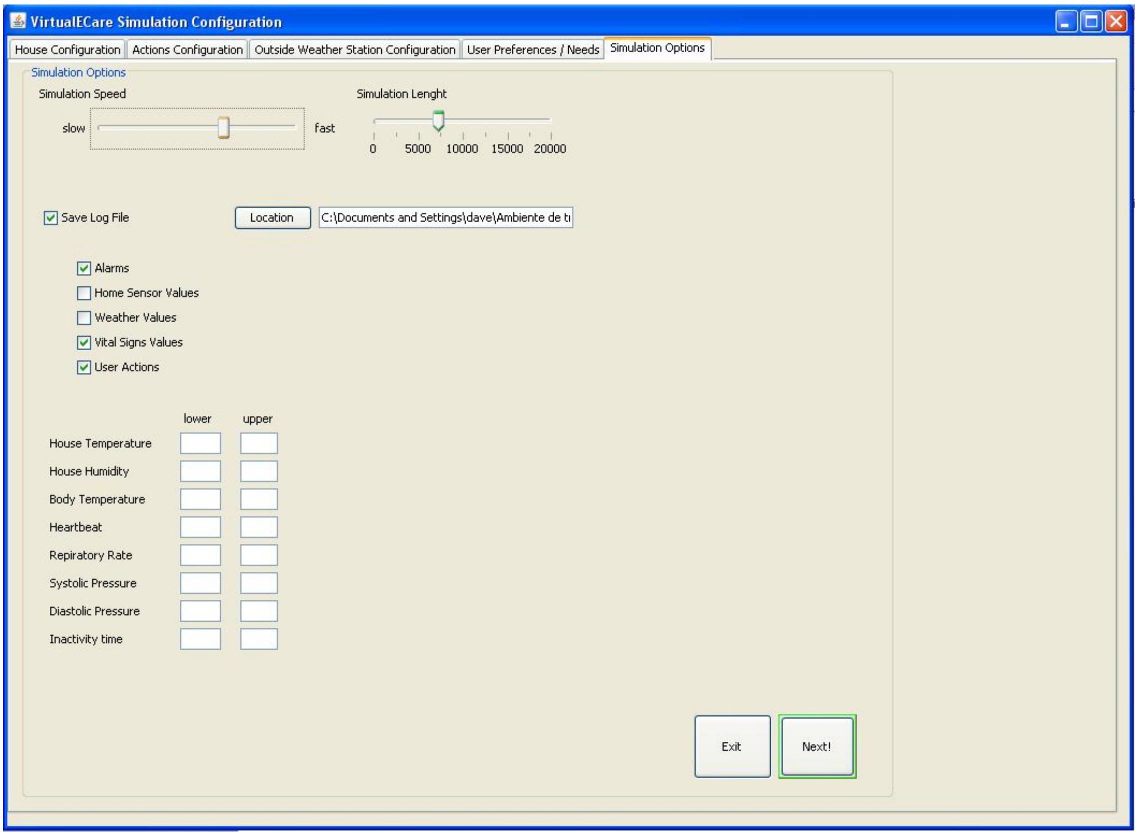

Fig. 5.5: The simulation options tab.

When the simulation starts, a bundle is responsible for controlling its execution. This bundle controls the length of each tick and this way, the speed of the simulation. At each tick, the control bundle asks all the other bundles of the simulation for all their values for the current tick and displays them on a GUI (Figure 5.7). It also provides a simple interface for interacting with the simulation (Figure 5.6) which enables the user to change the simulation related parameters.

Fig. 5.6: Control of the simulation.

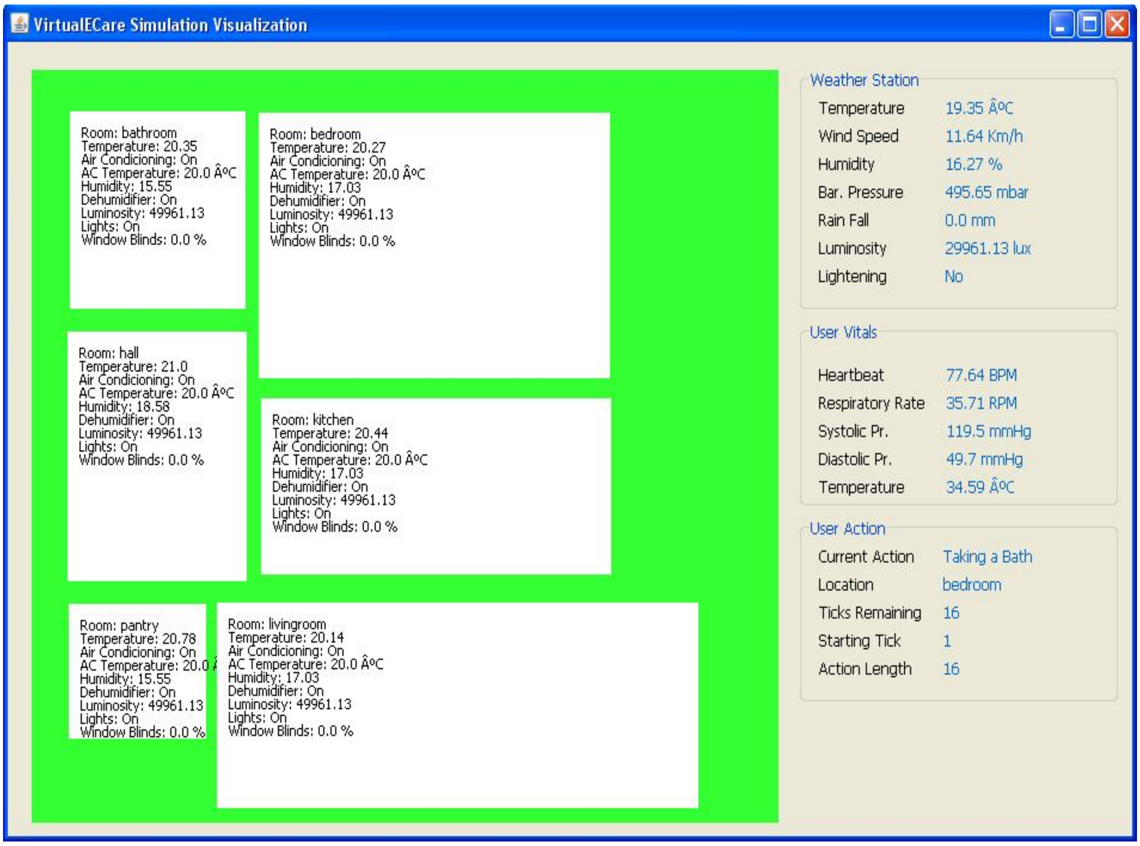

Fig. 5.7: The Simulation visualization.

5.6 Conclusion

Simulation is a widely used tool in many different scientific areas but its advantages have not been very exploited in the assisted environments field. In this work the decision was to develop a full simulation platform that could simulate all the main parameters associated with home environments, ranging from the user, his vital data and his interaction with the appliances to the house itself, its environmental parameters and the outside environment. With this simulation platform we were able to start the development of some reactive modules and test their behavior. We could also study how the results of the actions of these modules influenced the environment represented in the simulation. Also important was the test of the communications between the components of the VirtualECare architecture and the correctness of the transmission of the important data. This simulation platform also allowed us to continue the development without having to wait for the real equipment to be acquired, namely sensors and actuators. More than that, it allowed us to create specific scenarios that would hardly occur in reality or that we, for some reason, wanted to study. Simulation revealed itself as a tool that allowed us to improve our work and fasten the implementation phase, while at the same time, increased our confidence on the behavior of the final architecture.

6 Monitorization

We are now ready to describe the monitoring module. It was built and improved using the data provided by the simulation platform. However, it now uses data provided by some real 1-Wire sensors already acquired. The Monitoring component is composed of some main modules: the Database, the Sensor Manager, the Sensor Monitor, the Sensors and the decision making. Al these modules evidently are encapsulated inside OSGi bundles. Additionally, a GUI that allows for all the functionalities to be tested was developed.

This chapter can be roughly understood by looking at Figure 6.1. Using a wide range of sensors one interprets the user and its surroundings (i.e. its context). This may include information about the environment, the other persons present and the relations between them, the economical context, the historic context, geographic context, among others. This data is of utmost importance for describing the user and its environment. When interpreting it, one can infer knowledge, which is more than just data. Having knowledge one can assign meaning to the data and the numbers and letters start to mean something. At this point the system is ready to act according to the present knowledge and the objectives.

This is what happens in the work explained in this chapter. Data is acquired from the sensors, it is interpreted and actions are performed on the system according to the knowledge generated and the user preferences and needs.

This component not only works on its own as an independent monitoring and management module as it also provides its services to the exterior, in the form of OSGi services. This means that this component is passive of being used by higher level architectures, like the VirtualECare.

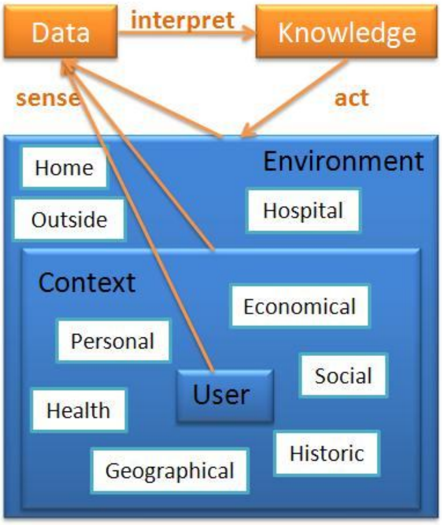

Fig. 6.1: The data flow in the monitorization module

6.1.1 Hardware

The sensors used in the project all 1-Wire sensors (Fig.6.1). Our 1-Wire network uses a DS9490R Adapter as a controller which connects to an USB port of the central computer. A telephone wire then spreads from this point to the house with sensors connected to it through RJ11 connectors. These connectors were used to make the configuration of the network easy. With this intention, the sensors were installed into small boxes with built-in connectors so installing a new sensor in a room is simply connecting a telephonic cable (Fig.6.2).

In our network there are two types of 1-Wire sensors: temperature and luminosity sensors. As temperature sensor the DS18B20 1-Wire Digital Thermometer is being used. It provides the temperature value with ±0.5°C accuracy over a -10°C to +85°C range (-67°F to +257°F). This sensor is therefore fit to work in a common home environment. The DS18B20 offers thermostatic functionality with over-temperature (TH) and under-temperature (TL) user-programmable setpoints stored in on-chip EEPROM. An internal flag is set when the measured temperature is greater than TH or less than TL. This feature can be used to check if since the last reset the temperature exceeded the limits established. Additionally, if thermostatic operation is not required, the two bytes of EEPROM reserved for TH and TL may be used for general-purpose nonvolatile storage. The DS18B20 is powered via a 3.0V to 5.5V power supply.

Fig. 6.2: A small 1-Wire temperature sensor.

As for the luminosity sensor, the chosen was the D2Photo Light Monitor (Figure 6.3). The D2Photo, light presence/absence monitor, provides the ability to monitor the state of fluorescent or incandescent lighting in a 1-Wire network. As the temperature sensor, this one has also an EEPROM that stores a variable that indicated if the state of the luminosity has already changed since the last reset. As the D2Photo supports conditional (Alarm) search which fastens the update of the information gathered, some functionalities can be implemented such as searching all the sensors on which the state of the light has changed since the last reset. This sensor operates on a temperature range of -40ºC to 85ºC and is powered via a 2.8 to 5.5 Volts power supply so it is also fit to work in a normal home environment.

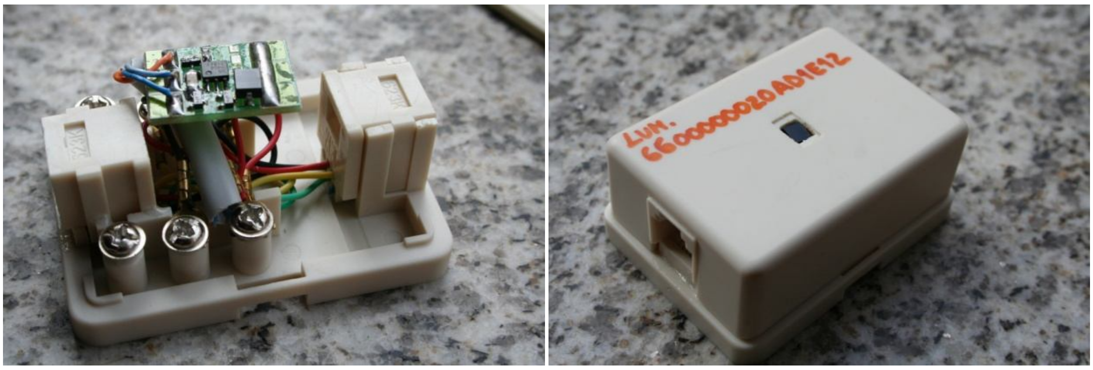

Fig. 6.3: Luminosity sensor installed inside a small plastic box with its address visible.

6.1.2 Database

The Database stores all the information of the sensors, namely it's unique address, the room it is in, the type of sensor, a brief description, among other. It also stores the information concerning the house and it's configuration as well as information about the user. All the operations or events that happen in the monitoring module (e.g. sensor added, temperature alarm fired) are also registered in the database, in a log.

The database is protected behind an OSGi bundle, which is the only way of interacting with it, granting the consistency and security, as described in the next section. Let us look now at the database structure.

The following tables were created in an Oracle database to implement this architecture (Figure 6.4). The table Rooms stores information about the rooms in the house: their names and their coordinates, as if they were designed in a Cartesian coordinate system. This coordinates will later be used by an extension of this work, to actually draw the house. The SensorsDesc simply associates a sensor address to a room, meaning that the sensor is in use at that room. if a sensor is not on this table, it means that it is currently not in use. As for the Sensors table, it contains all the sensors, whether they are currently in use or not, and their type and description.

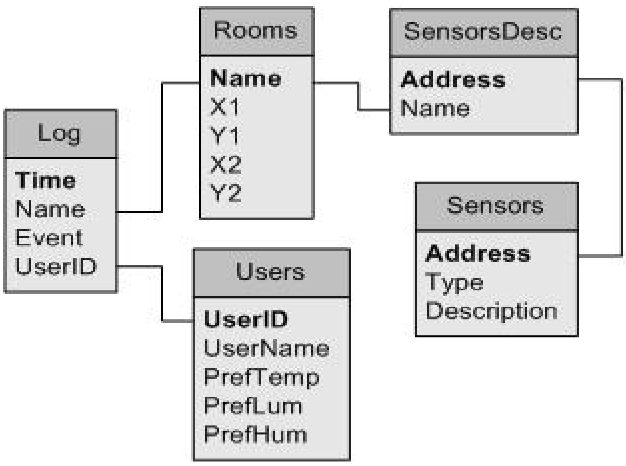

Fig. 6.4: The tables and their connections.

The table log contains the information about the events that take place in the environment, like the user or room involved, the time at which the event occurred and the description of the event. As for the Users table, it contains information about registered users, their ID number, and their preferences relatively to some environmental parameters.

6.1.3 Management

The Sensor Manager provides functionalities concerning the configuration of the sensors. Using the Sensor Manager one can associate a sensor to a room, move sensors between rooms, remove sensors from the rooms, add or remove sensors from the database or read detailed information about each sensor. It also constantly checks for new sensors in the network and issues warnings every time a new one is detected, as well as if some sensor in use in a room is not reachable. It also allows for conditional searches like searching for all alarming sensors. It acts as a bridge between the user and the database, automatically issuing all the important commands and making sure no invalid operations are executed.

This bundle is an important contribution to the 1-Wire technology as it transforms a regular 1-Wire network in a plug-and-play 1-Wire network. This adds a whole new range of possibilities. It makes it much easier to configure the 1-Wire network. One just has to lay down the cable and connect the sensors, which is also very easy since they were built into plastic boxes with telephonic connectors. As one goes on connecting new sensors, this bundle asks what to do with the new sensors, being possible to associate them to a room. It also detects when, by some reason, some sensor is registered as being in use but is not reachable, offering the possibility of removing it from the database.

6.1.4 Monitorization

As for the Sensor Monitor, it is responsible for providing the values of different sensors to the platform as OSGi services (for more details consult Appendix A – Monitorization Architecture). What this bundle does is to encapsulate all different types of sensors behind an OSGi bundle so that they are accessible to the other bundles. It knows how to communicate with each type of sensor and converts the information received from them into information that can be accessed by any other component of the system. This way, it is possible to hide the singularities of each type of device making it easier to develop applications that use 1-Wire devices. One does not even need to know how 1-Wire technology works nor how to communicate with sensors. Requesting a value is as simple as using an OSGi service.

This bundle also has some basic capacities for monitoring the values of the sensors. It constantly measures their values and issues warnings in case they are beyond predetermined values. Using this bundle it is also possible to interact with the sensors the other way around, i.e., to send data to them. For example, one can write in the sensor the Trigger Points which are the points at which the sensor enters in the alarming state or set the resolution of the sensor. It is also possible to reset the state of the sensors or to write some data in some types of sensors.

6.1.5 GUI

To make some tests and improve the architecture a GUI was created (Fig. 6.5). This GUI is an OSGi bundle that interacts with the Sensor Manager and Sensor Monitor bundles, using all the services provided. It is therefore possible to manage and monitor the sensors through this interface. One can select a sensor and view detailed information about its functionalities.

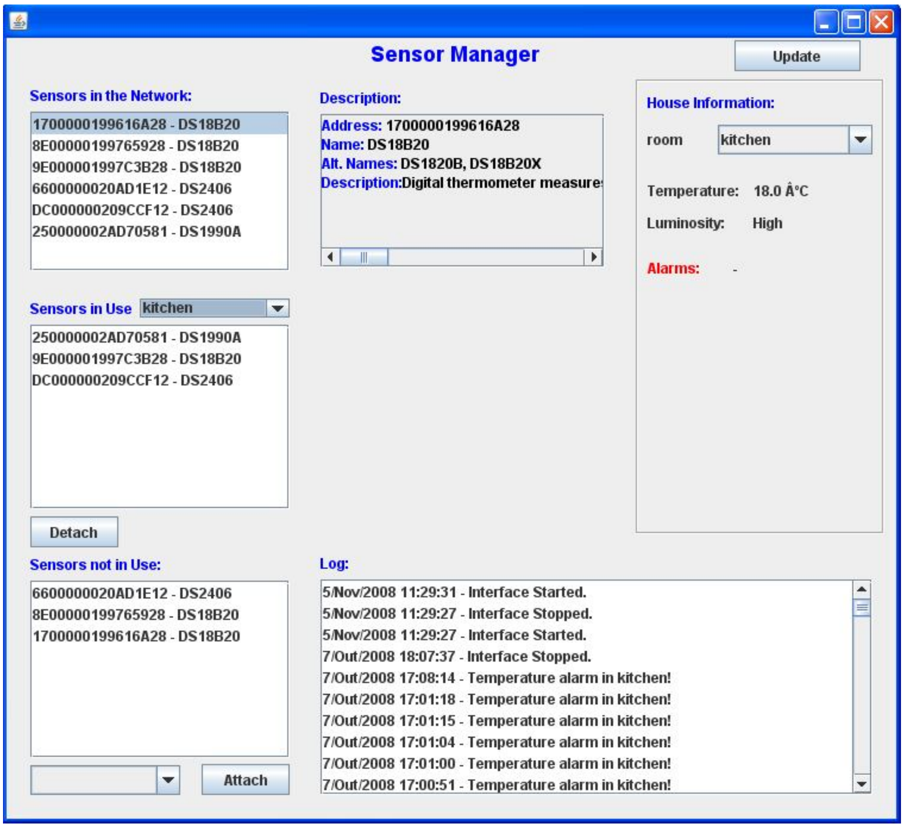

Fig. 6.5: The Sensor Manager Interface.

It is possible through this interface to associate the sensors to a room or see which rooms have which sensors. The values from the temperature and luminosity in each room retrieved from the sensors can also be seen here. Finally, this GUI also shows the alarms to the user: sensor unreachable, sensor alarming, new sensor added, temperature or luminosity alarm, among others. The information on the event log is also visible.

6.2 Home Actuators

As for the actuators, the X10 technology is being used. X10 is a standard for communication between electric devices. The devices supported by this technology are mainly oriented for the home automation area. It uses the power line for carrying the signal to the electric devices but it may also use its own wireless protocol. Other important features are its low price and its wide spread use. Using this technology, the person or the system can interact with appliances such as lights, dehumidifiers, air conditioning, among others from distance. To integrate the X10 technology with the rest of the architecture so that its devices could be used by any of its members, it was necessary to encapsulate the devices behind OSGi bundles. With that objective, an OSGi bundle was created that is able of interacting with X10 devices. What it basically does is to provide as services the commands that can be sent to each device it controls. Then, when a service is invoked, the bundle creates the equivalent X10 command and issues it trough the power line network.

The X10 commands are the way of communicating with the devices and encode specific actions: on, off, dim, brighten, all lights on and all lights off. These commands can be issued from different kinds of device. In our case, we can interact with the appliances in two ways: using the PR3130 wireless command or the CM15 computer interface.

The wireless command can be programmed and is especially useful for persons with movement limitations since they can interact with appliances without having to be near them. During the development phase it also had an important contribution which was to test the reception of the commands on the central computer. The CM15, in another hand, connects to the central computer through the USB port. In each case, the wireless commands are received by a transceiver module and then introduced into the power line. This standard allows the control of up to 256 devices which is enough in common houses.

For actually controlling the devices we are using two types of X10 modules: the PR2060 and the PR2025 (Figure 6.6). The first one is a transceiver that besides allowing the control of appliances receives the wireless commands and passes them to the power line. If an incandescent light bulb is connected to the PR2060, the intensity of the light can be controlled. It reacts to the commands On, Off, Dim, Bright, All Units/Lights Off, All Lights On. As for the PR2025, it enables the control of one light bulb and also reacts to the commands On, Off, All Units/Lights Off, All Lights On. Any of these devices is very simple to install being only necessary to insert the devices on the socket or on the E27 Edison screw base, respectively. This means that if the client wants to extend the system with more capabilities (e.g. control a new lamp recently bought) it is very easy to do it himself.

When using the PR3016, the interaction with the devices can be triggered locally or remotely. Locally it starts from the monitoring module, when some event happens and some action on the environment is needed. For example, if the temperature is high, the local monitoring module issues an X10 command to turn on the air conditioning device of the corresponding room. The X10 bundle however also provides the services remotely so that actions on the house can be performed remotely. The objective is that they can be used by members of the VirtualECare platform, namely the GroupDecision. As an example, the GroupDecision can turn on the lights on the house if it detects that the user is feeling bad during the night. Additionally, using these remote services, more features can be easily developed upon them, like interfaces for remotely interacting with the devices inside the house that can be used by the user.

Fig. 6.6: Some X10 modules used.

6.3 Outside Monitoring

As the user sometimes leaves its environment, this project was extended with some ability to monitor the user outside. Thru the use of a PDA build in GPS module, the system can know the location of the user at any time, as long as it has a network connection available. The ideal would be to have information about its vital signs as well. That, however, is at this time not possible.

Fig. 6.7: The output from the getLastKnownLocation service, visualized on Google Maps.

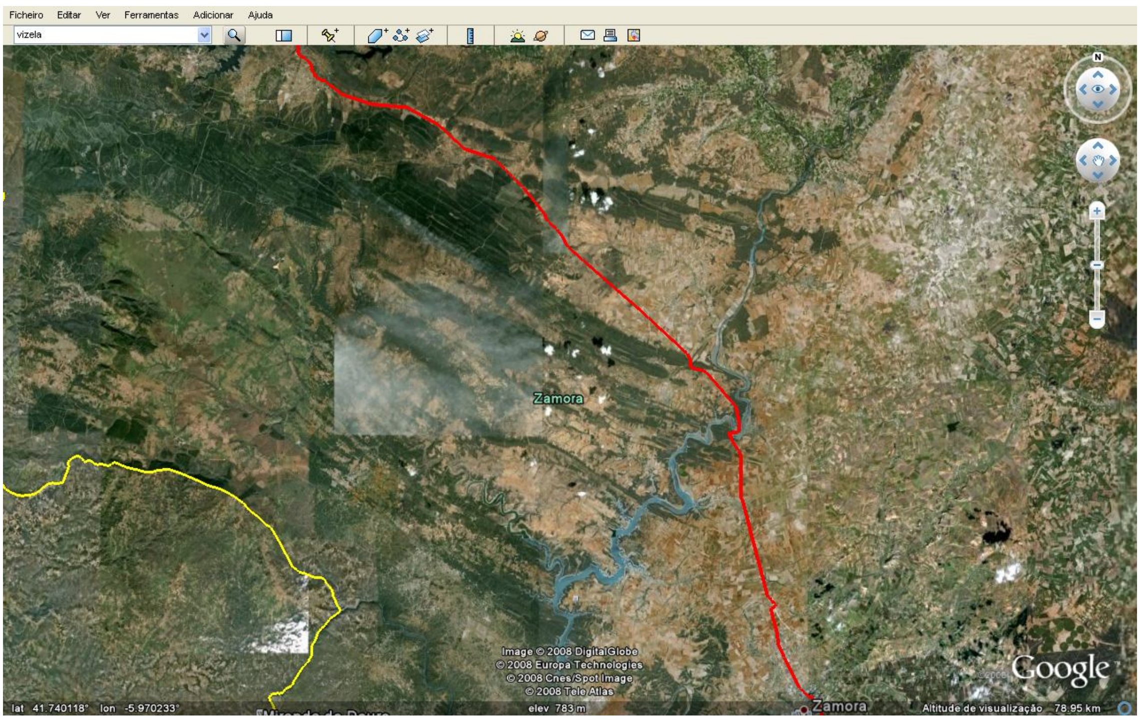

Fig. 6.8: The KML output file from the recordPath service, visualized on Google Earth. In this case, the client was travelling from Salamanca, Spain to Braga, Portugal.

The bundle responsible for tracking the user outside has some useful services for the user itself and for the system. The bundle records the user path (Figure 6.8) so that it can be later analysed in case of need and provides the last known location of the user (Figures 6.7).

It can also make the path from a given location to the last known location of a given user. This service is especially useful when there is the necessity to go rapidly to the user location, in cases of emergency. A similar service is also available to the user that consists on providing a path from its current location to a given destination (e.g. a hospital, a relative's house, etc) (Figure 6.9). The possible destinations are stored in a XML file and some other services are also provided like going to the nearest relative's home of the list or to the nearest location of type Hospital. All these services are provided using Google Maps API and Google Earth API.

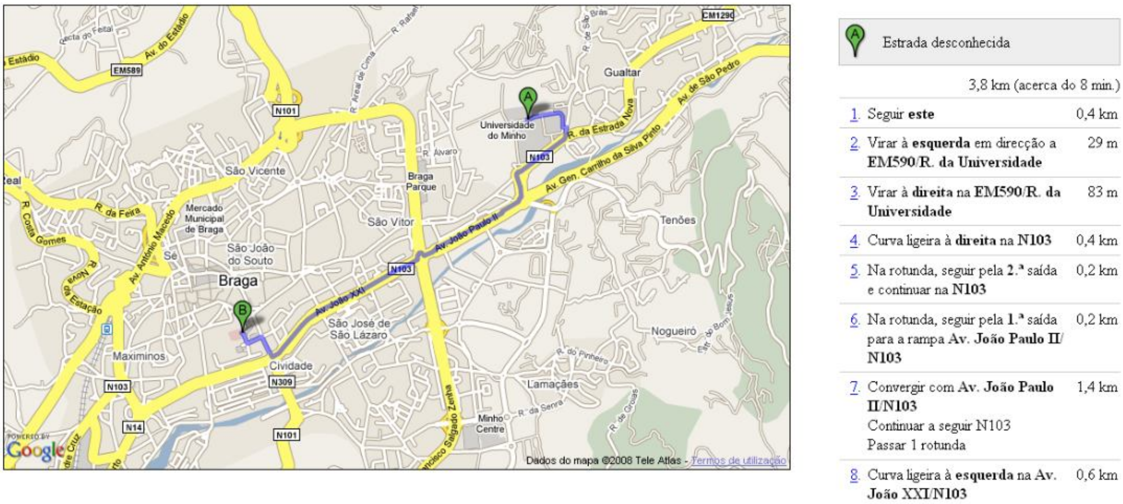

Fig. 6.9: The output from the GetPathTo service. In this case the client is in University of Minho and wants to go to the nearest hospital in the database which is located at lat: 41.547407, lon: -8.423090.

6.4 Autonomous Decision Making Through CBR

In an Ambient Assisted Living environment, there is a group of services that can be provided to the user. In the everyday life, the most common behaviour is that the user conscientiously engages in an interaction with the services in order to use them. The same happens in most of the projects that currently exist in the field of Ambient Intelligence. This can evidently be an obstacle to persons with movement impairments. What we propose in this work is that the services are automatically chosen by the system so that, in an optimum point, the user does not need to directly interact with the system and the environment is automatically managed like the user would manage it [65]. Of course that this is no simple task, mainly because each user is a unique case, with its own preferences and needs: a good decision in the context of a given user may not be so good in the context of another user [66].

The first important step to achieve the maximization of the user satisfaction through autonomous service choosing is to enrich each service with a strong description. It is not hard for us to decide between turning on the light and pulling up the window blind when the luminosity is low inside the house. But that is because we have our own description of the features and possible consequences of each action which we have gained through past experiences. We know that the lights are more effective but more energy consuming, while the window blinds are less energy consuming but may not be so effective on increasing the luminosity, depending on more factors like the size of the window or the external luminosity. In fact, when we make these kinds of choices, we are in a process of analyzing costs versus profits, although we do it so naturally that we do not even realize it. These analyzes are based on the description of the services that we have built in our mind and keep updating through our interaction.

6.4.1 Services description

In order to achieve a complete service description we must identify what are the main parameters that we take into account when we select a service in our own houses and do the same for each service in our system. Therefore we divided the description in two sets of properties: the functional properties and non-functional properties. The functional properties describe properties that relate directly to what the service does or the actions that will be performed when the service is requested. That is important to evaluate the possible consequences of invoking a given service. The non-functional properties describe constraints or principles that must be met in order for the service to be available.

We also defined a closed set of words that are enough to describe the services present in a common house. This set of words is evidently easily expansible if new services appear that need so. There words are organized in three groups:

- **Alarms** – These words identify an event in the room. It can be a fire alarm, a high or low temperature alarm, a high or low luminosity alarm, an intrusion alarm, the user pressing the panic button, among others.
- **Actions** – The words under this group identify the actions that a service can take. The actions can include calling the firefighters, calling a relative, turning on or off the heat or moving a window blind. These actions represent in fact methods that are automatically invoked when the service is requested. These methods encapsulate calls to services or devices of the environment.
- **Descriptions** – The words under this group give a meaning to the words that come next to them. As an example the word **uses** denotes that the service wants exclusivity over a given resource, the word **needs** denotes that the service needs to use a given resource, the word **against** denotes that the service is intended to counter a given alarm.

Let us look as some examples of services description. In the following code we can see the description of the air conditioning rising temperature service, the fire alarm service from the security bundle and a service from the window blind to control the temperature.

```
    name: tempup
    causes: tempup
    needs: power
    actions: Method(tempUp(String))
    level: health
    opposite: Method(tempDown(String))
    against: templow

    name: fire
    causes: water
    actions: Method(powerOff(), callAmbulance(),
callFirefighters(), callRelative())
    level: security
    uses: power
    against: fire

    name: wbtemp
    causes: templow, lumlow, lumhigh
    needs: power
    actions: Method(blindDown(String), blindUp(String)
    level: health
    uses: power
    against: temphigh
```

The services are organized in three categories according to its area of actuation: they are either security, health or comfort services. This categorization was the start of a solution to a problem that we run into while developing this work: the cycles. As an example, imagine there is a fire alarm. What the system does is to activate the sprinklers. This action of course doesn't take much time to activate the flood alarm. The system identifies the sprinkler as being the source of the flood alarm and turns the sprinkler off, which activates once again the fire alarm and so on.

The solution to these problems include a way of comparing the importance of the services (its categories), a way of comparing the services inside the same category and the previously mentioned description. Following the same example, the sprinkler and flood alarms are at the same category: security. However, we consider that a fire alarm is more important than a flood alarm since it can cause more damage more quickly. Therefore, a service against fire would have more priority than a service against water. At last, using the description the services indicate which resources to lock and which resources they need in order to work.

In this specific example, the fire alarm would request the lock of the power and the sprinkler and would obtain them even if they were locked by another service since the fire alarm is the service with higher priority. It would then activate the sprinkler. When the flood alarm went on, it would try to gain the control of the sprinkler but as it is locked by a service with higher priority, that would not happen and the service would be blocked by the system. With this technique we are able to solve these kinds of problems.

The priority of the services is as follows: Security > Health > Comfort and the priority of alarms is: fire > gas > intrusion > panic > water > temperature low > temperature high > humidity > luminosity low > luminosity high. These services are all local and therefore they do not need in their description references to their geographical location. Another parameter that is common in services description id their availability. This parameter is however not needed in our case since the system currently scans the services currently available and the ones that are not running are simply not detected. This is possible thanks to the previously mentioned features of the OSGi platform.

6.4.2 Learning to Choose Services

In the example mentioned previously, the selection of the services is not very hard to do since it involves basic decisions. In our everyday life, when we are faced with some situations that demand a choice that can dictate our survival, generally the mechanisms involved are very simple and of fast response. The same happens here.

However, in the home environments there are decisions that may involve much more variables. Retrieving the example of what to do when the luminosity drops, there are several factors that may influence the decision, namely: the power consumption of each service, the effectiveness, our willing to save energy or the amount of money that we are prepared to pay on the electricity bill, the external luminosity, the time until being night outside, among others. In our everyday life we make these decisions without even noticing that we are weighting all these parameters. However, a computer system needs to justify all its choices and must have all this into account. More problematic yet is that one may have the best optimization functions for a given user, according to its preferences or needs and the same functions, to another user, may be far from optimum. This means that it is useless to spend much effort improving some given optimization functions if they are suited only for a limited number of users. More than that, using optimization functions, the system would not evolve, would be static and would not adapt to the user. Imagine that the preferences or the needs of the user changed. In order for the system to correctly do its work there would be the need to study the new preferences of the user and then choose or develop a new set of optimization functions that would maximize its satisfaction. That is evidently not an effective solution.

Therefore we defend that such systems should not be static but instead adapt to the users. Supporting this idea, we developed this system so that it learns the preferences and habits of the user and, when it is sure enough that in a given case, the user would take a given action, the system itself takes the action instead of the user. In order to achieve this, we rely on a Case Based Reasoning [67, 53] model (Figure 6.10).

All the process is triggered by an alarm. As said before, an alarm can be any kind of event that occurs in the house such as temperature, luminosity, humidity, fire, intrusion or gas alarms. Every time an event happens in the house, the system creates a new case. This case can be seen as a snapshot of the state of the house in that moment. A complete case, as when it is stored in the knowledge base, contains the following information:

- Room – the room in which the alarm occurred;
- Alarm – the alarm which the case refers to;
- Service – the service used by the user to solve the alarm;
- Value – a value which denotes the success of this case when chosen by the system;
- Time – the instant of time at which the alarm occurred;
- Sensors – the information about all the sensors in the room and the weather station sensors.

When an event occurs, the system searches in the knowledge base for the case which contains the same alarm and that is more similar to the case just created. The measure of similarity is higher when the values from the sensors are closer and when the room is the same. After selecting the closest case, two things can occur: the case is close enough so that the action in the case can be taken by the system or the user interaction is requested.

The first case means that the system has a high confidence on its case and that the user would do the same if he had to choose. When it occurs the case may either be used if the similarity is high or it may be adapted. Adapting cases is relatively common and occurs when there are cases that are similar in the values of the sensors but occurred in different rooms. The adaptation occurs by creating a new case with all the parameters of the selected case but with the room where the alarm occurred. The new case is then added to the knowledge base.

Every time an action is automatically taken by the system a process starts in which the behaviour of the user is analyzed for a few moments. This process has as objective to determine if the action automatically taken by the system is correct. If, during a predetermined period of time, the user takes no action against the action that was automatically taken by the system, the value of the case increases. This indicates the system that the case is correct and will be preferred in future uses.

In another hand, if in that period of time the user takes an action that behavior must be studied. Two things can occur at this point. The user cancelling the action or the user cancelling the action and additionally executing another action which is similar to the action cancelled. If the user simply cancels the order the system issued, that means that the case is correct, i.e., the action performed was correct, what is wrong is the instant at which it occurred. This results in a slight change on the values on the respective alarm. As an example, if the high temperature alarm is activated and the system automatically turns on the air conditioning and the user right after turns it off, this means that maybe the user wants the air conditioning to turn on only when the temperature is a little higher. As a consequence, the limit of the temperature alarm for that room rises a little bit and the value of the case remains the same.

Maintaining the same example, if the user besides cancelling the system action takes another action that has the same objective that the cancelled action, it means that the time at which the alarm occurred is correct, what may not be correct is the action taken. A good example is the system turning on the lights because the low luminosity alarm was activated and, after that, the user turning the lights off and moving the window blind up. The effects of this happening are that the value of the existing case is decreased a little bit and a new case is added which contains the state of the house and the action taken.

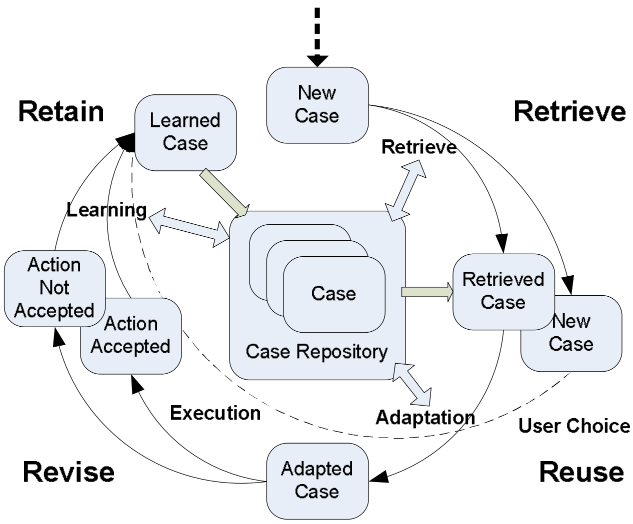

Fig. 6.10: The Case Based Reasoning model used, based on the one presented by [53].

When there are no cases that are similar enough and the user interaction is requested, the services that are available and suited to resolve the alarm are selected and shown to the user so that he can select. The services are presented to the user with minimal information, only with an intuitive image describing it. (Figure 6.11) This kind of interface is thought mainly to be used in devices with touch screens which makes the services selection a very easy process to the user, making the learning process easier for the system. When the user clicks on an image, the system adds the information about the action taken by the user to the new case and the case is added to the knowledge base now that it is complete. At last, the action is executed by the system, once again without the user having to move to where the device is and the system learned a bit more about the user preferences.

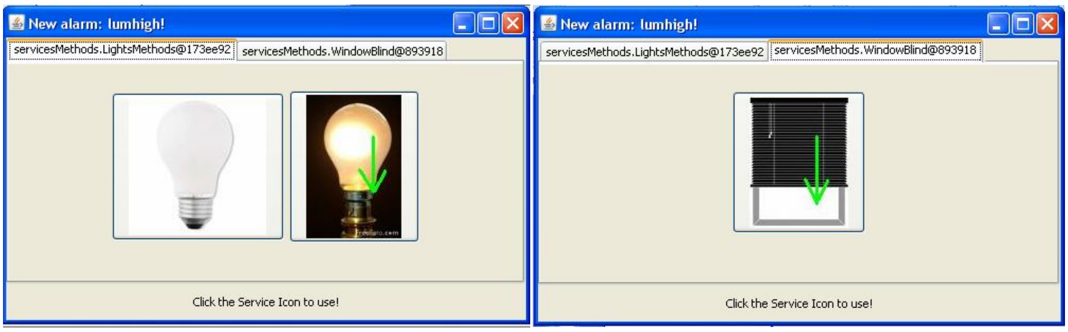

Fig. 6.11: The user can select among three services to deal with the high luminosity alarm.

This Case Based Reasoning model has many advantages for the user, being the most important that it is not static. It adapts to the user from the first moment, mimicking its actions. This means that after some time, each system will be unique and will be a mirror of its user. More than that, the system will keep evolving as the user interacts with the house, even without direct interaction between user and system. This means that even if the user preferences change, the system will adapt to it with the time, without any external interaction. That would not be possible if instead of using this method one used optimization formulas since they would have to be changed according to the preferences of the user.

After some time of interacting with the system what the user has is an entity that manages the house as he would manage it, taking actions that do not need its interaction anymore, releasing him from boring tasks that he would probably have difficulty in performing.

6.5 Conclusion

The monitorization of the environment and his inhabitants is one of the main concerns of AAL. Monitorization consists on the continuous providing of data about the entities being monitored or the providing of specific data when predetermined events occur. In this work we have contemplated both of these situations. In one hand, the environment is continuously monitored and the values obtained are logged so that later, if needed, they can be studied. The data obtained is also analyzed on real time so that if any parameters go beyond specific limits, actions are triggered. All the data collected has been used for the system to learn, based on a CBR methodology, how the user manages his environment with the objective of mimicking its behaviour in the system behaviour. The first steps on outside monitorization have also been taken with the integration of a GPS module into the architecture. This module is able of providing in real time the location of the person and the path travelled. The user can also use it with the objective of requesting directions to a predetermined set of locations such as hospitals, police stations or relative's houses. In the whole, this architecture has shown itself fit for monitorization based on very different devices and able of incorporate other to complete its functionalities.

7 Conclusions and Future Work

The generalized problem of the ageing population is maybe one of the most important challenges that we face in this new century. We know nowadays that, if changes aren't made, social security systems and health care institutions are likely to collapse due to the growing costs of the older population. If the need for changes is a certainty, the doubt may be how to change.

We believe that this change should be made recurring to the new IT technologies and to support this belief the work documented in this dissertation was done. These technologies are cheaper and more available and allow for things that only a few years ago seem impossible. Concepts like TelePresence, TeleConference, video calls and many others are nowadays common and are an example of how people can be virtually in more than a place at the same time, thus reducing transportation costs and others. These new technologies can be used by doctors to provide care and advise without leaving their office and with the person comfortably sitting on its environment. This is just a mere example of the possibilities raised by these technologies.

During the execution of these work we concluded that it is possible to build a monitorization and assistance environment for people in need that relies on cheap technologies and devices. These kinds of environments are in fact what we believe to be the future of basic health care. In this paradigm, people are kept in their own environments therefore reducing costs and increasing their comfort. Moreover, these systems can be adapted to be part of higher level architectures (like the VirtualECare project), resulting in profits for both parts and, evidently, for the user.

The objective of take a step ahead in this direction was pursued during the time of this work and the initial objectives mentioned in section 1.6.3 were accomplished, as described during the document. Having done this, we contributed with the suggestion of a possible path to solve one of the most important challenges that we are facing nowadays.

7.1 Synthesis of the Work Done

The main contributions of this work can be organized as follows:

- A multi-agent OSGi bundle. This bundle can interact with Jade agents and provides methods for starting or stopping them from the exterior. It is also possible to interact with the agents transparently, requesting their services like they were regular OSGi services. Agents inside this bundle can also use services offered by the other bundles in order to complete their tasks.
- Two 1-Wire OSGi bundles, one for interacting with temperature sensors of type DS18B20 and the other for interacting with luminosity sensors of type D2Photo. Each of these bundles offer services related to the sensors mentioned, exposing their functionalities to the other bundles (e.g. read temperature, read luminosity, write in the sensors memory). This bundle uses a DS9490R USB – RJ11 interface.
- A X10 bundle which is able of interacting with X10 appliances through a CM15 USB interface. This bundle exports services that enable for X10 commands to be sent to specific addresses.
- A Database bundle which interacts with an Oracle database. This bundle provides services that enable a basic set of commands to be sent to an instance of an oracle database.
- A GPS bundle that is able of interpreting NMEA strings acquired from a given COM port. This bundle provides the common services that we expect to use from a GPS such as

velocity, position, latitude, longitude, among others. Other bundles can therefore use this information just by invoking common services. The device used was the built-in GPS module of a HTC P3600.

- A proposal of a Multi-Agent based architecture that is able of supporting a multitude of different devices and technologies, providing a way of communication and interconnection. This architecture is also very modular, extensible and dynamic which enables for the continuing of its development and the inclusion of new technologies.
- A Simulation platform that simulates the main components of a house and its surrounding environment. The simulated values include internal and external parameters like temperature, luminosity, humidity or others, the user movement and actions with the corresponding effects on the environment and devices and appliances. As the simulation is built on OSGi technology, it is possible to interact with it so that external commands can be sent to the simulated devices, which respond in a realistic way. Everything in the simulation is fully configurable, from the design of the house, which also plays an important role on the evolution of the simulation to the behaviour of all the simulated values.
- A proposal for solving the known problem of choosing services in an autonomous and intelligent way, based on a CBR model. This proposal was developed and verified on the previously mentioned platform.

The work done represents an advance on the field of Ambient Assisted Living, providing means for the integration of common technologies on these kinds of environments. The solution adopted for selecting the services also represent a new way of solving this problem which no longer relies on inflexible optimization rules but instead shapes to the user as the user interacts with the house, resulting in a personalized and constantly evolving monitoring and assistance system.

7.2 Relevant Work

The work developed and documented in this dissertation is, as stated before, integrated in the VirtualECare project. This project is being developed at the Computer Science and

Technology Center in the Department of Informatics (DI-CCTC), at the University of Minho. This research project has been submitted to FCT (Foundation for Science and Technology) under the reference *PTDC/SAU-ESA/103755/2008 - VirtualECare - An Ambient Assisted Living* and is now awaiting approval.

Part of the work presented in this dissertation has been awarded with the **TLeIA08** – a National Award for Artificial Intelligence projects. This award is attributed by the Portuguese Artificial Intelligence Association and is intended to reward master student's projects in the fields of Artificial Intelligence. The awarded work titled "Virtual Assisted Living Environment" focuses on the development of simulation platforms for Ambient Assisted Living.

The progress of this work was documented on the following publications:

Journals:

- Carneiro D., Novais P., Costa R. and Neves J.: Simulation Environments in AAL. (Submitted to IEEE Intelligent Systems journal, 2009).
- Novais P., Costa R., Carneiro D., Marques A., Machado J. and Neves J.: An Assisted Living Environment. (Submitted to Journal of Ambient Intelligence and Smart Environments, 2009).

Conferences with peer-review:

- Carneiro D., Novais P., Costa R. and Neves J.: Case-Based Reasoning Decision Making in Ambient Assisted Living, in IWAAL - International Workshop of Ambient Assisted Living, Part II, LNCS 5518, University of Salamanca, Spain, 10-12th June, 2009 (to appear).
- Costa R., Novais P., Lima L., Carneiro D., Samico D., Oliveira J., Machado J. and Neves J., VirtualECare: Intelligent Assisted Living, in Electronic Healthcare, Dasun Weerasinghe (ed.), Springer-Verlag, Series: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, (Revised Selected Papers of The 1st International Conference on Electronic Healthcare in The 21st Century, City University, London, England, in September 8 - 9, 2008), pp 138-144, ISBN978-3-642-00412-4, 2009.
- Carneiro D., Costa R., Novais P., Neves J., Machado J., Neves J., Simulating and Monitoring Ambient Assisted Living, in Proceedings of the ESM 2008 - The 22nd annual

European Simulation and Modelling Conference, Le Havre, France, October, ISBN 978-90-77381-44-1, pp 175-182, 2008. (Indexed by ISI Web of Science)

- Costa R., Carneiro D., Novais P., Lima L., Machado J., Marques A., Neves J., Ambient Assisted Living, in Advances in Soft Computing, Vol. 51, Springer- Verlag, ISBN 978 978-3-540-85866-9, pp. 86-94, 2008 (3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008 (UCAMI 2008),Salamanca, Spain, 22-24 October 2008). (Indexed by ISI Web of Science)
- Novais P., Costa R., Carneiro D., Machado J., Lima L., Neves J., Group Support in Collaborative Networks Organizations for Ambient Assisted Living, in Towards Sustainable Society on Ubiquitous Networks, Makoto Oya, Ryuya Uda, Chizuko Yasunobu (eds), Springer-Verlag, Series: IFIP International Federation for Information Processing, ISBN 978-0-387-85690-2, pp 353-362, 2008 (The 8th IFIP Conference on e-Business, e-Services, and e-Society (I3E 2008), Tokyo, Japan, 24-26 September 2008). (Indexed by ISI Web of Science)

In the course of the work developed under the VirtualECare project, I have been present in the 3rd Symposium of Ubiquitous Computing and Ambient Intelligence and in the International Symposium on Distributed Computing and Artificial Intelligence both held in the University of Salamanca, Spain in 2008. Also in the scope of this work, I will attend the International Workshop of Ambient Assisted Living held in the University of Salamanca, Spain from 10 to 12 June, 2009.

7.3 Future Work

Although the objectives of the work have been accomplished, some work has yet to be done to transform this into the desired reality, being maybe the most important to implement this architecture in a real environment, with people in need of care and study the feedback. In order to do that, a list of further tasks was identified:

- Acquire new sensors to measure more parameters in order to maximize the information about the context of the house and about the user vital signs. As more information about the context is obtained, the decisions will be more wisely taken by the system and the personal using it.

- Acquire RFID equipment to identify multiple persons inside the house and to locate them. Extending this system to identify multiple persons is a major step, opening new doors as its implementation in environments like hospitals or care centres.
- Improve the decision making mechanism to support multiple persons and to deal with the possible contradictions of trying to satisfy the needs of more than one person.
- Implement the system in a real multi-person environment and study the acceptance of the persons involved (caregivers, patients and relatives) as well as the effective changes in the routines of these three groups of persons.
- Keep studying new technologies that appear every day and its viability to integrate the system, as well as the possible advantages.

Bibliography

[1] Weil, D.: Population Aging. Department of Economics, Brown University and NBER, 2006.

[2] Turner, D. et al.: The Macroeconomic Implications of Ageing in a Global Context. OECD Economics Department Working Papers, No. 193, OECD Publishing, 1998.

[3] United Nations. World Population Ageing: 1950-2050. UN, 2002.

[4] Malmberg, A. Industrial Geography: agglomeration and local milieu. In *Progress in Human Geography*, vol. 20, 392-403, 1996.

[5] World Health Organization: Active Ageing a Policy Framework. WHO, 2002.

[6] Dosi, G.: Sources, procedures and microeconomics effects of innovation. In *Economic Literature*, 26, 1120-1171, 1998.

[7] Al-Hamad, A., Flowerdew, R., Hayes, L.: Migration of elderly people to join existing households: some evidence from the 1991 Household Sample of Anonymised Records. In *Environment and Planning* A 29(7) 1243 – 1255, 1997.

[8] Statistics Portugal. Estatísticas Demográficas 2006. INE, 2006.

[9] Novais P., Costa R., Carneiro D., Machado J., Lima L., Neves J.: Group Support in Collaborative Networks Organizations for Ambient Assisted Living. In *Towards Sustainable Society on Ubiquitous Networks*, Makoto Oya, Ryuya Uda, Chizuko Yasunobu (eds), Springer-Verlag, Series: IFIP International Federation for Information Processing, 2008.

[10] Costa R., Novais P., Machado J., Alberto C., Neves J.: Inter-organization Cooperation for Care of the Elderly, In *Integration and Innovation Orient to E-Society*. Wang W., Li Y, Duan Z., Yan L., Li H., Yang X., (Eds), *Springer-Verlag,* Series: IFIP International Federation for Information Processing, ISBN: 978-0-387-75493-2. 2007.

[11] Costa R., Novais P., Lima L., Carneiro D., Samico D., Oliveira J., Machado J. and Neves J.: VirtualECare: Intelligent Assisted Living. In *Electronic Healthcare*, Dasun Weerasinghe (ed.), Springer-Verlag, Series Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp 138-144, 2009.

[12] The Amigo Project: Amigo – Ambient Intelligence for the networked home environment. Short project description. 2004.

[13] MIT Artificial Intelligence Laboratory: Oxygen – Pervasive, Human-Centered Computing. Available at http://www.oxygen.lcs.mit.edu/Publications/Oxygen.pdf.

[14] Haigh K., Kiff L., Myers J., Guralnik V., Krichbaum K., Phelps J., Plocher T., Toms D.: The Independent LifeStyle Assistant: Lessons Learned. Honeywell Laboratories, 2003.

[15] Feldman A., Tapia E., Sadi S., Maes P., Schmandt C.: ReachMedia: On-the-move interaction with everyday object. In *Wearable Computers.* Proceedings of the ninth IEEE International Symposium. Ambient Intelligence Group, MIT Media Laboratory, 2005.

[16] Camarinha-Matos L., Afsarmanesh H.: Virtual Communities and Elderly Support. In *Advances in Automation, Multimedia and Video Systems, and Modern Computer Science,* pp. 279-284, 2001.

[17] B. Somekh: Action Research: a methodology for change and development, Open University Press, 2005.

[18] IST Advisory Group: Ambient Intelligence: from vision to reality.
[19] IST Advisory Group: Scenarios for Ambient Intelligence in 2010; Final Report, 2001.
[20] Weiser M., Gold R., Brown J. S.: The origins of ubiquitous computing research at PARC in the late 1980s. IBM SYSTEMS JOURNAL, vol. 38, No. 4, 1999.
[21] Vasilakos, A., Pedrycz, W.: Ambient Intelligence, Wireless Networking, And Ubiquitous Computing. Artech House Publishers, 2006.
[22] Riva G., Vatalaro, F., Davide, F., Alcaniz, M.: Ambient Intelligence: The Evolution Of Technology, Communication And Cognition Towards The Future Of Human-Computer Interaction. IOS Press, 2005.
[23] Rocha M., Cortez P., Neves J.: Adaptive Learning in Changing Environments. In Verleysen M. (Ed.): *Proceedings of 11th European Symposium on Artificial Neural Networks* (ESANN'2003), Bruges, Belgium, pp. 487-492, 2003.
[25] Bick M., Kummer T.: Ambient Intelligence and Ubiquitous Computing. In Handbook on Information Technologies for Education and Training, part I, subpart 1, pp. 79-100. Springer Berlin Heidelberg, 2008.
[24] Rubel P. et al.: New Paradigms in Telemedicine: Ambient Intelligence, Wearable, Pervasive and Personalized. In *Wearable eHealth Systems for Personalized Health Management: State of the Art and Future Challenges*. IOS Press, 2004.
[26] Acampora G., Loia V., Nappi M., Ricciardi S.: Ambient intelligence framework for context aware adaptive application. In *Computer Architecture for Machine Perception*. CAMP 2005. Proceedings of the seventh International Workshop, Volume 4, Issue 6, pp. 327 – 332, 2005.
[27] Juan C. Augusto and Paul McCullagh: Ambient Intelligence: Concepts and Applications. Invited Paper by the International Journal on Computer Science and Information Systems, volume 4, Number 1, pp. 1-28, June 2007.
[28] Jorge J.: Adaptive tools for the elderly: new devices to cope with age-induced cognitive disabilities. In *Workshop on Universal Accessibility of Ubiquitous Computing,* Proceedings of the 2001 EC/NSF workshop on Universal accessibility of ubiquitous computing: providing for the elderly. ACM, 2001.
[29] Fink J., Kobsa A., Nill A.: Adaptable and Adaptive Information Access for All Users,
Including the Disabled and the Elderly. In *User Modeling: Proceedings of the Sixth International Conference UM97, pp. 171-173*. Springer, 1997.
[30] Starner T. et al: Augmented Reality Through Wearable Computing. Presence, Special Issue on Augmented Reality, 1997.
[31] Badami V., Chbat N.: Home Appliances Get Smart. Spectrum, IEEE, vol. 35, No. 8, pp. 36-43, 1998.
[32] Perrey, R.. Lycett, M.: Service-oriented architecture. In *Applications and the Internet* Workshops, Proceedings. vol., no., pp. 116-119, pp. 27-31, 2003.
[33] Costa R., Neves J., Novais P., Machado J., Lima L. and Alberto C.: Intelligent Mixed Reality for the Creation of Ambient Assisted Living. In *Progress in Artificial Intelligence*, Neves J., Santos M. and Machado J. (eds), Lecture Notes in Artificial Intelligence 4874 Springer, ISBN 978-3-540-77000-8, 2007.
[34] Costa R., Carneiro D., Novais P., Lima L., Machado J., Marques A., Neves J.: Ambient Assisted Living. In *Advances in Soft Computing*, Vol. 51, Springer- Verlag, pp. 86-94, 2008.
[35] Perng J.K., Fisher B., Hollar S., Pister K.S.J.: Acceleration sensing glove (ASG). In *Wearable Computers*, 1999. Digest of Papers. pp.178-180, 1999.
[36] Talukder A., Yavagal, R.: Mobile Computing. McGraw-Hill Professional, 2006.

[37] Satyanarayanan M.: Fundamental challenges in mobile computing. In *Principles of Distributed Computing*, Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing, 1996.

[38] Chess D., Grosof B., Harrison C., Levine D., Parris C., Tsudik G.: Itinerant Agents for Mobile Computing. In *IEEE Personal Communications*, October 1995.

[39] Rao B., Minakakis L.: Evolution of mobile location-based services. In *Mobile computing opportunities and challenges SPECIAL ISSUE: Mobile commerce opportunities and challenges*, pp. 61 – 65, 2003.

[40] Ball M., Lillis J.: E-health: transforming the physician/patient relationship. In *International Journal of Medical Informatics* 61, pp. 1-10, 2001.

[41] Istepanian R., Laxminarayan S., Pattichis C. (Eds): M-Health: Emerging Mobile Health Systems. Springer, 2005.

[42] Marreiros G., Novais P., Machado J., Ramos C., Neves J.: An Agent-based Approach to Group Decision Simulation using Argumentation. In *Proceedings of the International MultiConference on Computer Science and Information Technology*, Workshop Agent-Based Computing III, Wisla, Poland, pp 225-232, 2006.

[43] Open Services Gateway Initiative: *Osgi Service Platform*, Release 3. IOS Press, 2003.

[44] García-Macías J., Avilés-Lopes, E.: Developing Ubiquitous Applications through Service-Oriented Abstractions. In *Advances in Soft Computing*, 3rd Sysmposium of Ubiquitous Computing and Ambient Intelligence, Springer, 2008.

[45] Riva, G.: Ambient Intelligence in Health Care. In *Cyberpsychology & Behavior*, vol. 6, 295 – 301, 2003.

[46] Weiser, M.: The Computer for the 21st Century. In *Scientific American* 265, Nr. 3, S. 94-101, 1991.

[47] Mota R., Vidal R., Alves V., Machado J., Abelha A., Nelas L.: E-learning in medical environments using intelligent tutoring systems. In *International Conference on Knowledge Engineering and Decision Support*. Oporto, 2004.

[48] Wooldrige M.: An Introduction to Multiagent Systems. John Wiley & Sons, 2002.

[49] Olson, G. M., Malone, T. W., and Smith, J. B. (Eds.): Coordination Theory and Collaboration Technology. Mahwah, NJ: Erlbaum, 2001.

[50] Brito L., Novais P., Neves J. The logic behind negotiation : from pre-argument reasoning to argument-based negotiation. In *PLEKHANOVA*, V., ed. lit. - "Intelligent agent software engineering ", pp. 137-159. London : Idea Group Publishing, 2003.

[51] Dey A.: Understanding and Using Context. In *Personal Ubiquitous Computing* 5 (1): pp. 4-7. 2003.

[52] Mitchell T.: Machine Learning. McGraw-Hill Higher Education, 1997.

[53] A. Aamodt and E. Plaza.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Communications, 7(1):39-59,1994. IOS Press.

[54] Kolodner, J. L.: An introduction to case-based reasoning. Artificial Intelligence Review, 6(1), 3-34, 1992.

[55] Kolodner, J.: Case-Based Reasoning. Morgan Kaufmann, 1993.

[56] Brachman R., Levesque H.: Knowledge Representation and Reasoning. Morgan Kaufmann, 2004.

[57] Active. ActiveECG web page. Available at www.activecenter.com. <Accessed on August 2008.>

[58] Paradiso R., Belloc C., Loriga G., Taccini N.: WEATHY - Wearable Healthcare Systems, New Frontiers of e-Textile. In *Personalised Health Management Systems: The Integration of Innovative Sensing, Textile, Information and Communication Technologies,* pp. 9-16. IOS Press, 2005.

[59] O.S.G.i: OSGi Service Platform, Release 3. IOS Press, 2003.

[60] Chen, K., and Gong, L.: Programming Open Service Gateways with Java Embedded Server Technology. Prentice Hall, 2001.

[61] FIPA: FIPA ACL Message Structure Specification. Available at http://www.fipa.org/specs/fipa00061/SC00061G.html. <accessed in 13-1-2009.>

[62] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services Concepts, Architectures and Applications, Data-Centric Systems and Applications Series. Springer, 2004.

[63] FIPA: FIPA ACL Message Representation in XML Specification. Available at http://www.fipa.org/specs/fipa00071/SC00071E.html. <accessed in 13-1-2009.>

[64] Carneiro D., Costa R., Novais P., Machado J., Neves J.: Simulating and Monitoring Ambient Assisted Living. In *Proceedings of the ESM 2008 - The 22nd annual European Simulation and Modelling Conference*, 2008.

[65] Carneiro D., Novais P., Costa R. and Neves J.: Case-Based Reasoning Decision Making in Ambient Assisted Living, in S. Omatu et al. (Eds.): IWANN 2009, Part II, Springer-Verlag LNCS 5518, pp. 787–794, 2009 (proceedings of the IWAAL - International Workshop of Ambient Assisted Living, University of Salamanca, Spain, 10-12th June, 2009) (to appear).

[66] J.C. Augusto, P. McCullagh, V. McClelland, and J-A. Walkden.: Enhanced Healthcare Provision Through Assisted Decision-Making in a Smart Home Environment. In *Proceedings of the 2nd Workshop on Artificial Intelligence Techniques for Ambient Intelligence* (AITAmI'07), pp. 27–32. IJCAI, 2007.

[67] Neves, J.: A Logic Interpreter to Handle Time and Negation in Logic Data Bases. In *ACM: The Fifth Generation Challenge* 50-54, 1984.

Appendix A

Simulation Architecture

This appendix depicts in more detail the architecture and the design of the system. Although it does not fully details these aspects due to their complexity, it provides a deeper insight on the conception of the system. In pictures A.2 to A.4, we will see the architecture of the simulation and monitorization platforms in terms of the OSGi bundles they are composed of. In these pictures, the bundles are represented by circles with their corresponding names and services are represented by triangles. As services can either be used or provided by bundles, bundles that provide a given service connect to the vortex of the triangle while bundles that use a service connect to the opposite side, as we can see on figure A.1.

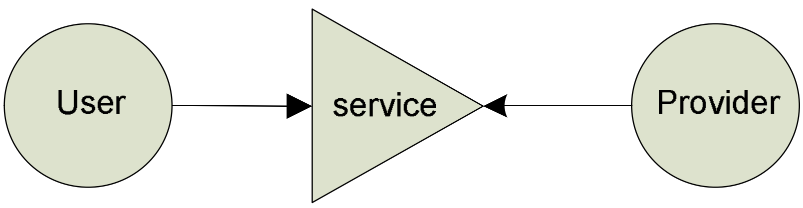

Fig A.1: Example of two bundles exchanging information by means of services.

In figures A.2 and A.3, it is possible to see all the bundles that are part of the simulation platform and the way they share information. Let us now move to a brief description of each bundle and the relations that are established between them:

- **StartGUI** – this bundle is responsible for starting all the simulation process. Its visible face is the GUI that is used for the configuration of the simulation process. Through that GUI one can set all the values and scenarios. These values are collected from the GUI and passed to all the necessary bundles. Examples of this are the *Weather*, *Vitals* or *Actions* bundles which receive the values that basically contain their running mode for that simulation instantiation. When the configuration is complete and the simulation is ready to start, the *StartGUI* bundle passes the values and starts the necessary bundles for the simulation to go on.
- **Weather** – the *weather* bundle is responsible for generating all the simulated data about the outside weather station. As it is the base for the generation of the remaining environmental parameters, this bundle uses no services from other bundles. It does however provide a wide range of services which can be grouped in two main groups. The first ones are used mainly by the *StartGUI* bundle, for setting all the configuration of the weather simulation, when the process starts. The second ones are the services for providing the simulated data and that can be used by the other bundles. As examples, the *SimulationInterface* bundle uses these services for showing the values on the interface and the *temperature* bundle uses the temperature service as one of the parameters for generating the internal temperature of each room of the environment.
- **Vitals** – the *vitals* bundle is very similar to the *weather* bundle, except it generates data that represents the user vital signs. It also provides a large number of services, some of them for configuring the vital signs generation and the others for providing the generated values. It does however uses a service from the *Actions* bundle as the *Vitals* bundle needs to know the activity of the user for a more realistic simulation of the effects of daily actions on the user vital signs.
- **House** – this bundle simulates the house and, for doing so, does not need to use any service from the other bundles. It provides services for configuring the house when the simulation is on the first stage, through the *StartGUI* bundle. It also provides, during the simulation process, the parameters about the configuration of the house so that other bundles can have access to them. These parameters include the data about the room’s insulation or connections and are used by the majority of the bundles that simulate some environmental parameter.

- **Luminosity** – this is a relatively simple bundle which provides as main service the value of the room's luminosity. For obtaining this value, the bundle uses services from the *House*, *Weather*, *Lights* and *WBlinds* bundles.
- **Humidity** – similarly to the *luminosity* bundle, this bundle simply generates the value of the humidity in each of the rooms and provides that value as a service. However, due to the intrinsic characteristics of the humidity parameter, this bundle uses services from the *Weather*, *House*, *Dehumidifier* and *Actions* bundles.
- **Lights** – the *lights* bundle simulates the lights inside each room. Besides the services for adding and configuring the lights, it also provides as service the state of each light and the possibility to change that state by changing the lights intensity or by turning them on or off. As reading the state of the lights can be useful for several bundles as the *Luminosity* one, only the *Reactive* bundle changes the state of the lights.
- **AC** – the AC bundle simulates the air conditioning devices in the house. It therefore allows for its state to be changed by the other bundles as well as its state to be read, mainly by the *Temperature* bundle which uses the air conditioning temperature as an important factor for simulating the temperature inside the rooms.
- **WBlinds** – this bundle simulates the window blinds of the house. It is also a very simple bundle which allows for the position of each blind to be read which is mainly useful for the *Temperature* and *Luminosity* bundles. The *Reactive* bundle can also set the position of the blinds through this bundle's services.
- **Reactive** – the *Reactive* bundle is one of the main complex bundles and it is explained in more detail ahead. It does not provide any service although it uses many services from the other bundles. Its main task is to constantly monitor the state of the environment and their components, including the user, and take appropriate measures in case of specific situations (e.g. temperature high, luminosity low). It also incorporates a simple case-based reasoning model that allows it to learn from the user how to use the services present in the house for managing the environment.
- **Simulator** – this bundle receives the control of the simulation from the *StartGUI* bundle when the configuration process ends. From that point on, it is responsible for controlling the pace at which new values are generated. It provides therefore services for setting the parameters of the simulation (e.g. length, tick length) which are used by the *StartGUI* when the simulation is about to start. Besides that, this bundle is also

responsible for reading all the values that are presented in the simulation interface and passing them to the *SimulationInterface* bundle, updating it.

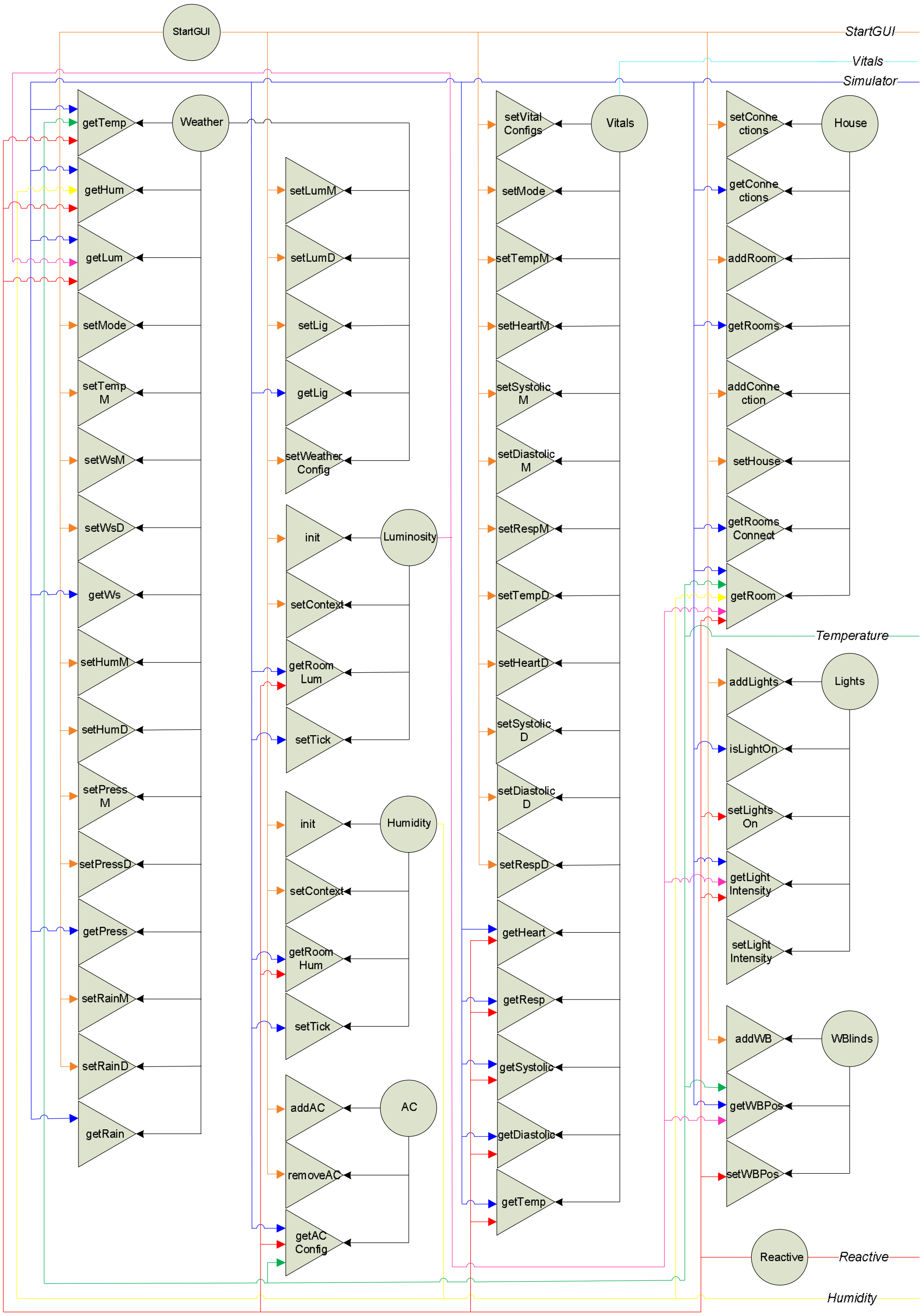

Fig A.2: First part of the simulation architecture from the OSGi point of view.

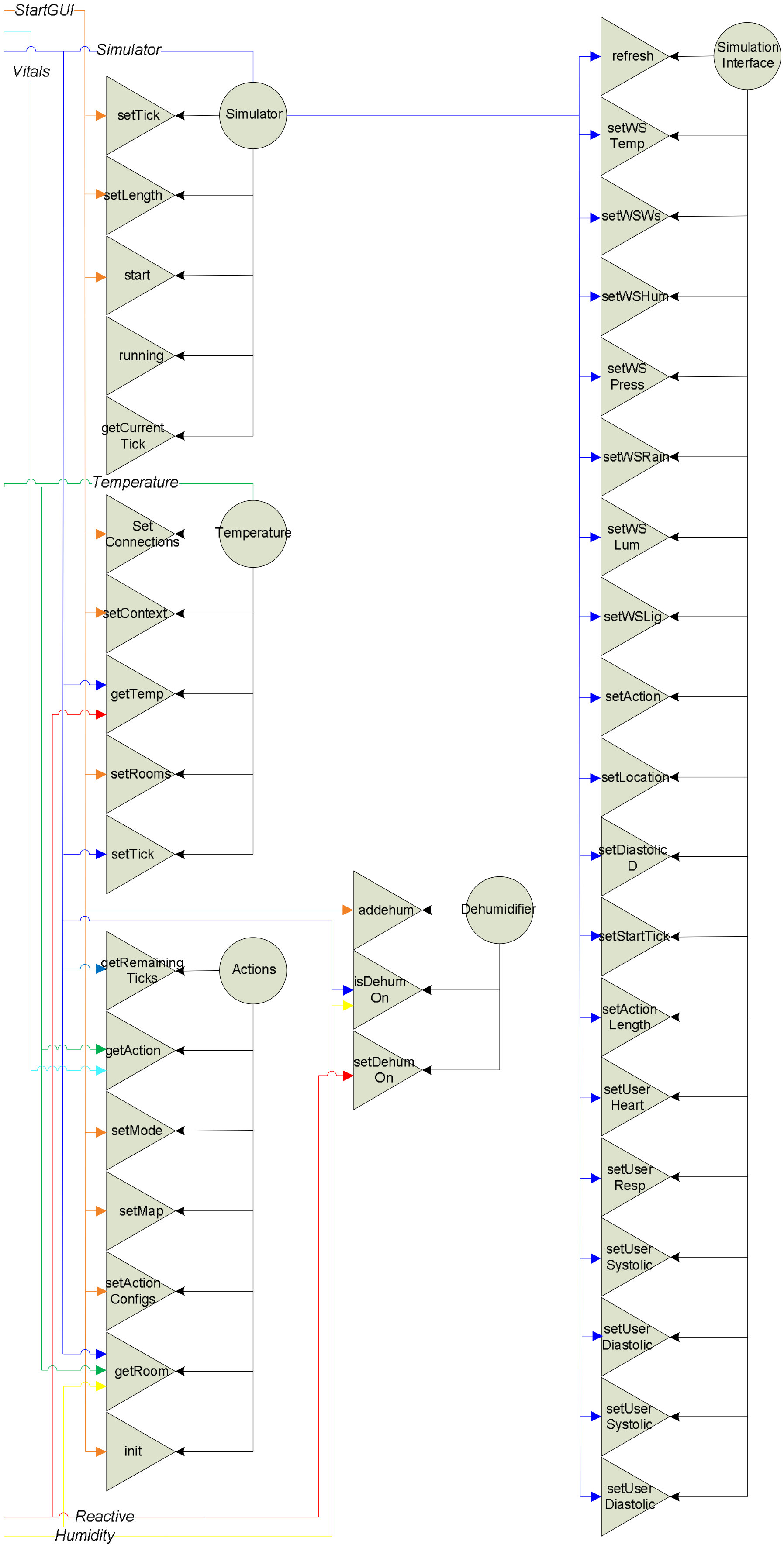

Fig A.3: Second part of the simulation architecture from the OSGi point of view.

- **Temperature** – the *Temperature* bundle simulates the temperature inside the several rooms of the house, providing the simulated values as a service. For generating these values it uses services from bundles like the *WBlinds*, *AC* or *Weather.*
- **Actions** – this bundle simulates the actions of the user. It provides some configuration services which are used by the *StartGUI* in the first phase of the process for setting the actions simulation mode (i.e. random, bounded random or planned) or for setting the rooms. Additionally, it provides as services the location and current action of the user as well as the number of ticks remaining for the completion of the current action. This data is mainly used by the *Temperature*, *Humidity* and *Vitals* bundles as the actions performed by the user influence the values generated by these bundles.
- **Dehumidifier** – the *Dehumidifier* bundle simulates a dehumidifier and therefore provides services for turning the dehumidifiers on the house on or off, being the effects of these actions visible on the simulated environmental parameters.
- **SimulationInterface** – the last bundle described here is a simple interface which shows all the important values of the simulation ranging from the house configuration, to the sensors values, the state of house appliances, the user actions, between others. All this data is passed by the *Simulator* bundle thus updating the interface.

The Reactive Bundle

In this section the *Reactive* bundle will be analyzed in more detail in terms of a Finite State Machine. In the following table the states and the transitions between them are depicted, accompanied by a brief description. The initial state is the *ScanParam* which constantly repeats scanning all the values of the sensors.

Table A.1: The states and transitions between them in the *Reactive* bundle.

Initial State	Transition	Final State	Description
ScanParam	Param>Hlevel	Alarming	The value of some sensor is read and if the value is higher than its corresponding predetermined level, the systems enters in alarming state.
	Param<=Hlevel	ScanParam2	If the value of the sensor is not higher than the high alarm level for that parameter, it must be checked if the value is lower than its low level.
Alarming	AlrmLstHas	ScanParam	If the new alarm is already on the alarm list then it is ignored and the system returns to the initial state
	AlrmLstHasOppAlrm	RemAlarm	If there is an opposite alarm to the new alarm in the alarms list the previous alarm is outdated and must be removed. The system enters in the RemAlarm state.
	AlrmLstHasN	NewAlarm	If the list of alarms has does not contain the new alarm, the alarm must be triggered and the system enters the NewAlarm state.
RemAlarm	ReleaseResources	ScanParam	When an alarm is removed, the information must be deleted from the alarm list and the resources (if any) used by the alarm must be released.
			Continued on next page

Continued from previous page

ScanParam2	Param>Llevel	CheckAlrms	If the level of the parameter is higher than the low level this means that the parameter is not alarming.
	Param<=Llevel	Alarming	If the level of the parameter is lower than the low level the system enters into alarming state.
CheckAlrms	releaseResources	ScanParam	If there was a previous alarm on the parameter that was being scanned the alarm must be removed and the resources released since the alarm is now outdated.
	NoAlarm	ScanParam	The value of the sensor is between the limits and there is no previous alarm so everything is normal.
NewAlarm	NoServicesAgainst	ScanParam	If there are no services against the alarm that has been created, the system cannot deal with it at this moment and returns to the initial state.
	HasServicesAgainst	NewAlarm2	If there are services that can be used agains the new alarm, the process goes on to the NewAlarm2 phase.
NewAlarm2	NeedsResourcesInUse	ResInUse	If the service needs to use a resource that is already been used, the services will need to be analyzed and decide which one will have the resource.
	NeedsRedourceNotInUse	Decision	If there is no other service using the resources needed for this service to run, the service takes control of the resources.
ResInUse	UsedByLowServ	TakeControl	If the resources generating the conflict are being used by a service with lower level, the new service takes control of the resources.

Continued on next page

Continued from previous page			
ResInUse	UsedByHighServ	ScanParam	If the resources generating conflict are being used by a service with higher level, the service cannot be used to counter the new alarm at this moment and the system returns to the initial state.
Decision	SingleService	TakeControl	If there is a single service suited for countering the alarm, the system chooses that service.
	MultipleServices	Decision2	If there are multiple services for countering the alarm, the system must choose the best one.
Decision2	SimilarCase	TakeControl	If there is a case that is similar enough with a previous known case with the same service, the service is chosen to counter the new alarm.
	UserChoice	TakeControl	If there is no similar case that can be used, the user has to make that choice and choose among the several services available, enriching the knowledge base.
TakeControl	TakeAction	ScanParam	When the service is selected, it is given control of the needed resources and the action or actions of that resource are executed. The system returns to the initial state.

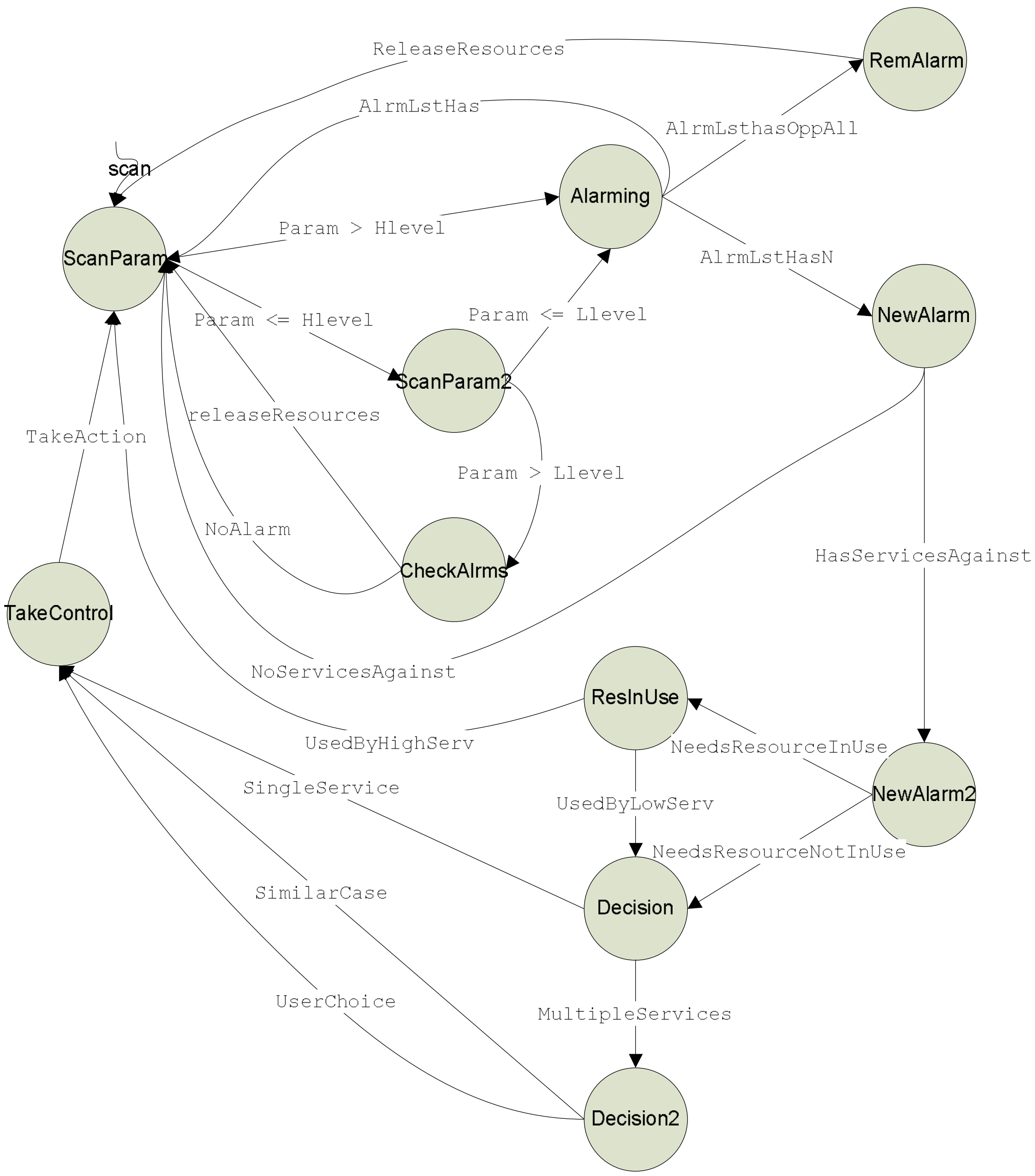

Fig A.4: The Finite State Machine diagram of the *Reactive* bundle.

Monitorization Architecture

Let us now move to a closer description of the monitorization bundle. This bundle puts together the components needed to monitor and interpret the environmental parameters of the house. It is therefore the result of merging together a group of very different technologies which includes multi-agent systems, databases and sensors. In figure A.5 the bundles and the services provided by each of them are represented. There are however only represented in the picture the services used by the own platform to perform its tasks. The services which appear as non-used are used by external applications to perform their tasks. Examples of these external applications are the VirtualECare platform and the Sensor Manager interface, which was developed as a way of testing all the functionalities of the bundles.

Let us now describe each of the four bundles of the platform:

- **MAS** – the *MAS* bundle is responsible for allowing Jade agents to "live" inside the OSGi platform. The services provided allow for agents and containers to be started and stopped as well as the Jade platform itself to be stopped. The agents created in the ambit of the monitorization platform have as main tasks to constantly monitor the values read from the sensors and fire alarms in given cases, showing a basic reactive behavior.
- **SensorManager** – this bundle provides services for sensor management being some of them are used by the *MAS* bundle. The main services are for adding sensors to the database and moving the sensors between rooms. This bundle in fact hides the database and at the same time ensures the integrity of the information stored, transforming the SQL statements in OSGi services.
- **1-Wire** – this bundle has the ability to interact with 1-Wire sensors, in this specific case with temperature and humidity sensors. The services provided are therefore in the ambit of reading values from sensors with given addresses and in sending information for the sensors, like setting the resolution or trigger points. The *MAS* bundle uses services from this bundle for reading the values of the sensors and the remaining services are used by the external applications.
- **SensorMonitor** – the S*ensorMonitor* bundle provides basic monitorization functionalities as getting all the alarming sensors or setting the points at which the

sensors enter in alarming mode. Only a few of these services are used by the *MAS* bundle and the rest of them is used by the external applications.

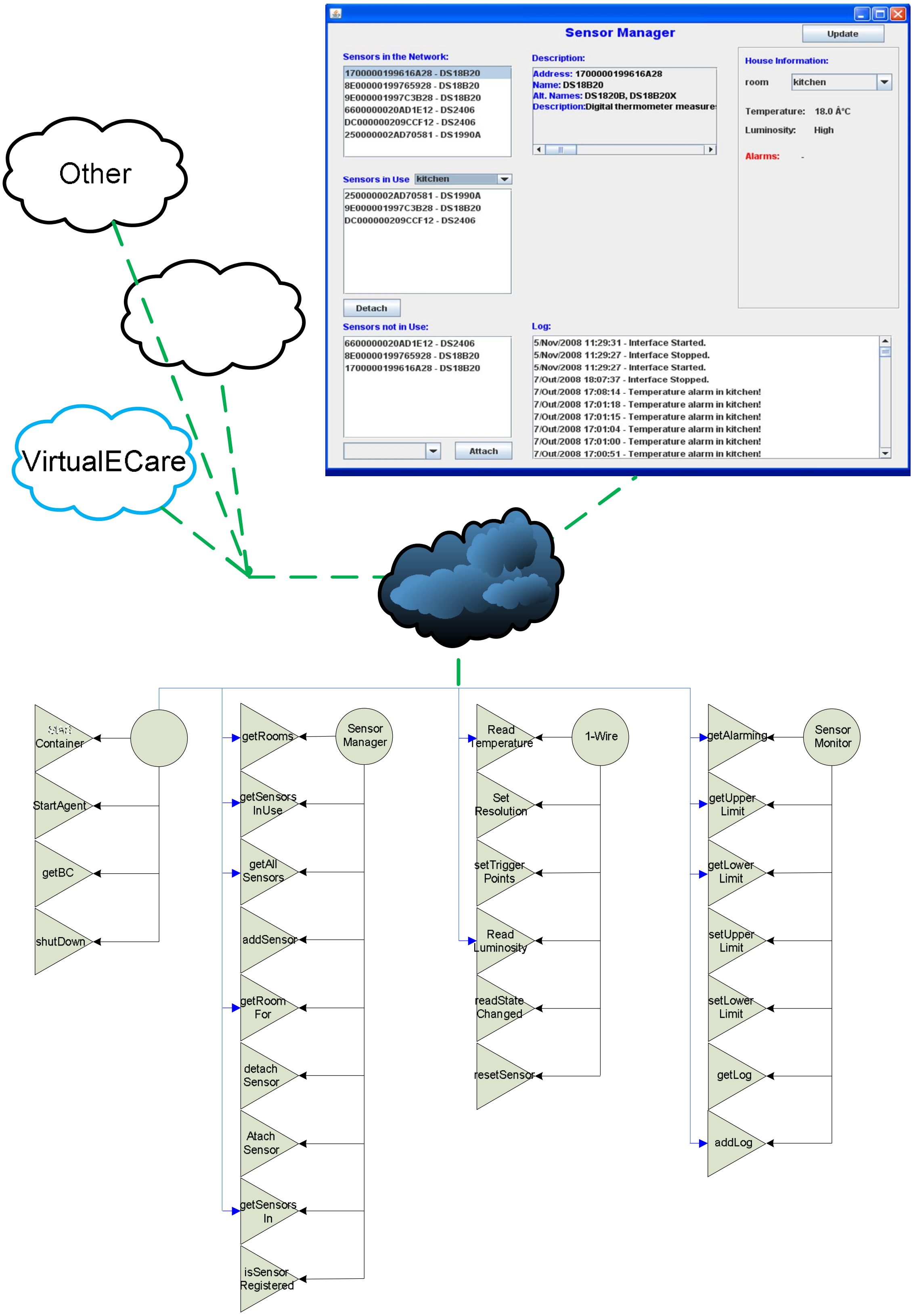

Fig A.5: The architecture of the monitorization platform from the OSGi point of view.