Ambient Assisted Living with Linux

Willy Allègre, Cédric Séguin, Thomas Burger, Florent de Lamotte, Pascal Berruet, Jean-Luc Philippe, Jean-Philippe Diguet

RenPar'20 / SympA'14 / **CFSE 8**
Saint-Malo, France, du 10 au 13 mai 2011

Ambient Assisted Living with Linux

Willy Allègre, Cédric Seguin, Thomas Burger, Florent De Lamotte, Pascal Berruet, Jean-Luc Philippe, Jean-Philippe Diguet*

Université de Bretagne-Sud
Lab-STICC - Centre de recherche
Rue de Saint Maudé - BP 92116
56321 Lorient Cedex, France.

Contact: `first.last@univ-ubs.fr`

Abstract
Population ageing is set to affect European countries over the coming decades, increasing the number of dependent people. In this context, Ambient Assisted Living (AAL) is one solution to enable these people to stay in their preferred environment longer, thus delaying hospitalization. To take up these challenges, this paper proposes an original linux-based solution for home automation developed at the Lab-STICC Laboratory. A hardware architecture that can be deployed in such an environment is first defined. To fit both hardware specification and user needs with home automation services, a linux-based software application has been developed. Thanks to the contribution and sharing of the Linux community, it is possible to save development time and customize your own application. This is crucial when it comes to the fast deployment of ad hoc embedded systems.

Keywords: ambient assisted living, embedded Linux, home automation, computer aided design

---

1. Introduction

1.1. Social background

People with disabilities sometimes have considerable difficulties, or even physical incapacities, to perform daily tasks on their own. Whether they are disabled physically, intellectually, or due to sensory impairment, their limited capabilities do not allow them to control a large range of home devices. Home automation is one possible solution to overcome these disabilities; it can enable them to do daily tasks without the systematic assistance of caregivers or relatives. This solution, which extends the time people can live in their preferred environment, answers one of the major social and economic issues on global population ageing [1]. Technological change can now respond to these demographic trends.

The drastic and continuous reduction of the cost, size, and consumption, coupled with the increased performance of electronic systems makes it possible to consider a new paradigm in the field of home automation. Thus, we no longer consider a monolithic application, centralized and based exclusively on a home automation bus, but take into account several "smart devices" able to process information and offer their own services: we focus on *Ambient Assisted Living (AAL)*, and more specifically, on *Pervasive Systems* dedicated to people with disabilities.

As part of ambient assisted living, original needs have emerged:

- Home automation and multimedia services should be flexible to fit user needs, which are closely related to their disability, but also to the configuration of their living space.
- While the majority of home automation solutions are limited to individual use, it is now necessary to address multiple-user environments (e.g. healthcare center) as well.

* This work is funded by the Bretagne regional council and the Morbihan local council

- Dependability and reconfiguration in case of failure represents the major challenges that are to be faced to ensure the delivery of a service (*QoS* ensuring [2]).

- Finally, the cost of the installation, the ease of deployment, and the intrusiveness of the solution are key criteria involved in the acceptance of home automation.

All these points naturally influence the decisions in terms of choice of hardware, communication protocol, and software architecture, leading to a tailored home automation solution.

1.2. Technological background

Adopting a home automation solution is not an easy choice due to the possible offers which are numerous, versatile, and usually expensive. On one hand, the home automation solution can be partial, offering, for example, lighting control [3], a security system [4], or a multi room ambient system for video and audio [5]. On the other hand, it can be a complete solution [6,7] which allows one to regulate the desired temperature, to manage home security, and to supervise the entire house. However, despite their usefulness, these home automation solutions are expensive (i.e. using commercial applications) and closed to one specific home automation technology. A pervasive system is composed of many heterogeneous devices (communication mode/protocol). This kind of system must be able to ensure interoperability between these devices, but also facilitates the integration of upcoming technologies.

Open source and free solutions, especially based on Linux [8], present good alternatives to proprietary ones. Nevertheless, the latter are not completely adapted to our specific domain. For a large-scale deployment, application has to be distributed on low-power cards with limited memory capabilities. Considering *LinuxMCE*, each *core PC*'s hard drive should have at least 4 GB to install the OS, which is far from our lightweight application needs. Even if home automation functionalities are well integrated, these kinds of solutions are above all media center applications with heavy functionnalities. Furthermore, due to these irrelevant features, the user interface is not adapted to people with disabilities (impaired vision, physical disabilities, etc.). Finally, all open or closed contributions need experts to configure a home automation solution. In the context of AAL, needs in terms of adaptation are constantly evolving. For financial reasons, the systematic intervention of an expert cannot be considered.

To take up these challenges, QuatrA was the first AAL solution for a home care system proposing both environmental control and wheelchair navigation (section 2). This project was the starting point of our research works, firstly followed by Danah, that added an open middleware based on an original service architecture (section 3). Nowadays, projects resulting from these previous works are still taking place at the Lab-STICC Laboratory. The objectives are twofold : i) simplify design to integrate non-expert designers and ii) ease the deployment of an AAL solution in a distributed environment. The design tools and methods, introduced in section 4, lead to a flexible and adaptive solution. This paper is first an overview of past projects, and then opens onto current works still in progress.

2. Hardware architecture

AAL architecture that has to be deployed in the living space must cover the whole environment, on the scale of a room, as well as that of a healthcare center, and thus control all the electronic devices. Besides, heterogeneous devices require being controlled with a single interface which can manage heterogeneous technologies. Furthermore, there can be many users asking for services from the same resource at the same time: the home automation installation should be shared, dealing with simultaneous access and conflict management.

QuatrA was the first project resulting from collaboration between the Lab-STICC Laboratory and Kerpape MFRRC [3]. Aware of this AAL paradigm, the QuatrA project focused on increasing the autonomy of the disabled, especially wheelchair-bound people [9]. The objective is two fold: i) assist people in wheelchair navigation while ii) managing electrical devices (beds, lights, doors, etc.), as well as media ones (television, cd player, etc.). The user can, for example, ask for the "*bedtime*" scenario and his

[2] Quality of Service perceived by the user [2]

[3] Kerpape Mutualistic Functional Reeducation and Rehabilitation Center: `http://www.kerpape.mutualite56.fr/`

wheelchair will drive him into the bedroom, activating all the needed services such as "*open the door*" and "*turn on the light*", to complete the objective.

Hardware architecture is defined to satisfy these needs (cf figure 1), characterizing different hardware devices, which embed a Linux-based operating system:

- The **user terminal**, embedded with the wheelchair-bound user, deals with interactions between the user and his environment. It is responsible for service publishing (via N800 PDA) and for sending commands to the central server via a mobile terminal (bluetooth connexion). The N800 runs on Maemo-based Tablet OS with OMAP2420 processor, one of the first ARM11-based SoCs.
- The **mobile terminal**, also embedded in the wheelchair, acts as a bridge between the user terminal and the central server via a stationary terminal (bluetooth connexion). It is also used for path calculation and wheelchair control using embedded sensors. For optimal performance and consumption, a Linux system is built from scratch, embedded in a x86 architecture(PCM9373 board).
- The **stationary terminal** is not only a bridge as mentioned above, but also a control unit for local environment (infrared and KNX/EIB[4] connexion). This non-embedded terminal is based on a classic Debian-based distribution (light Ubuntu without desktop environment) implemented in both x86 and ARM processor boards.
- The **central server** receives all user requests, processes, and routes them according to the order and the rights of requesting users. It is also used for handling navigation or service conflicts, reconfiguration, and path calculation. This server is also based on a generic Debian distribution (Ubuntu) coupled with a Gnome desktop environment to provide a graphical user interface.

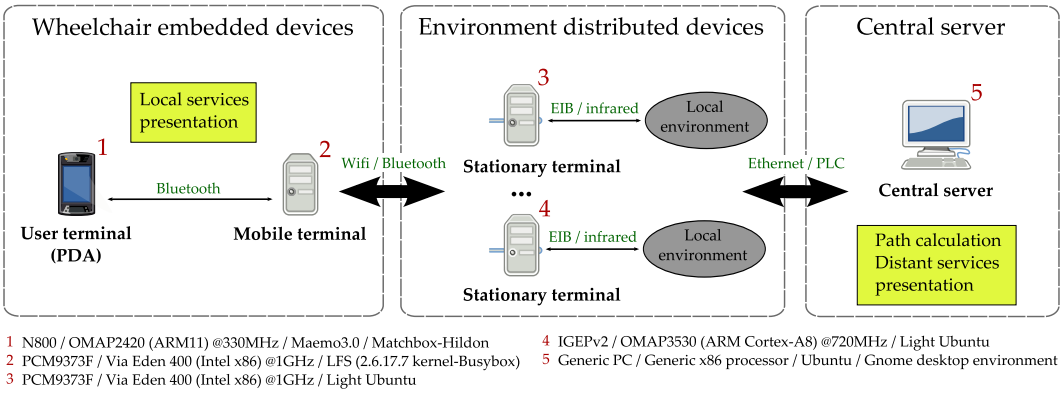

Figure 1: General hardware architecture from Quatra project

Consider the "*bedtime*" scenario described above. The user, far away from his bedroom, activates this service via his user terminal (bluetooth connected to the embedded mobile terminal). The command is then sent to the central server via stationary terminals. After path calculation, the latter sends the individual commands to stationary and mobile terminals. Once the itinerary is received, the wheelchair starts to move toward its goal. If on the way, a door is closed, the local stationary terminal automatically opens it using an EIB bus. However, if the door stays closed, the system will try to fix the failure, for example, by changing the itinerary, using central server calculation capacity. Once in the room, the stationary terminal sends a command to start the radio (infrared communication), switches on the light, and closes the bedroom door (home automation bus communication).

Current research works is thus based on the above-presented hardware architecture. However, this architecture model is not closed and makes it possible to evolve. Nowadays, progress has been made with embedded Linux for ARM processors, which is why the trend is to change mobile and stationary

[4] The KNX/EIB (European Installation Bus), born of a consortium, is an open standard for home and building control.

terminals first based on Pc/x86 (*PCM 9373*) to ARM technology (*IGEPv2*) with more development opportunities, a smaller size, and energy saving possibilities (cf table 1).

Board	Processor	DSP	Power	PCB size
PCM 9373F	VIA Eden 400 (Intel x86) @1GHz	none	2.40A; 5V	5.7"x4"
IGEPv2	OMAP3530 (ARM Cortex-A8) @720MHz	TMS320DM64+ @430MHz	0.65A; 5V	3.6"x2.5"

Table 1: Comparative specification of investigated boards

Although there are different systems according to terminal specifications, the same Linux execution environment guarantees interoperability among heterogeneous platforms, abstracting peripheral devices.

3. Software architecture

The software part is built considering the hardware architecture and specification described above. First, starting "from scratch" allows one to better fit hardware specifications, to select features during building configuration, and to develop functionalities dedicated to AAL. To reduce time and development costs, open source and free software are used for the design of a home automation application. This is made possible by the large number of developers from the Linux Community who create and maintain these open source and free software packages.

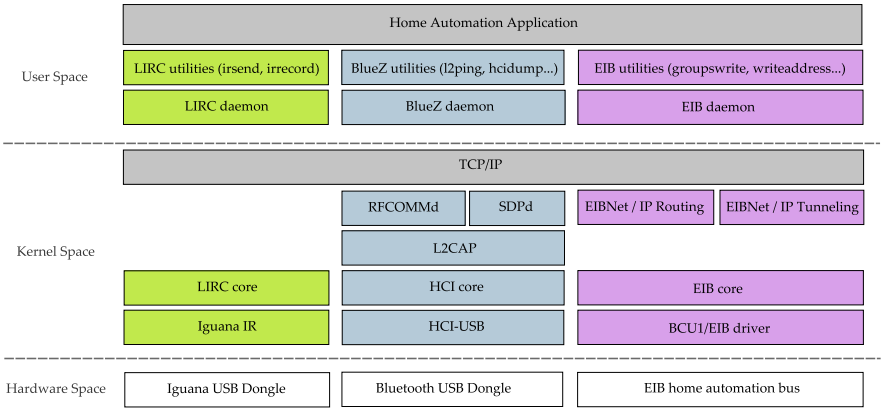

Figure 2: Software architecture for a GNU/Linux based Home Automation Application

The software architecture, which has to be embedded in user, stationary, and mobile terminals (cf section 2), has the same structure, no matter the hardware target. It is then possible to add and remove communication protocol packages or drivers to suit the specific terminal. Figure 2 gives a part of this software architecture. First, the *BlueZ* [5] stack is used for all wireless communication, requiring no long distances or high speeds. This stack, installed by default in all Linux distributions, is used by opening and managing bluetooth sockets in a C program, in the manner of IP sockets. Then, the *eibd* daemon from the BCU SDK [6] allows one to control a KNX/EIB installation. After configuring the physical addresses of all *eib bcus*, it is possible to control such a bus from a stationary terminal application. Finally, Infrared stands for the communication standard for multimedia devices that can be found throughout the house. *LIRC* [7], used with the *iguanaIR* driver and its USB dongle [8], is then used to control these infrared devices.

Focusing on both environmental control, as well as navigation, the Danah project integrates all or part of these packages and drivers to build home automation middleware [11]. The server application, which

[5] The official Linux bluetooth stack that provides support for core bluetooth layers and protocols: `http://www.bluez.org/`
[6] Bus Coupling Unit Software Development Kit by the Automation Systems Group of the *Technische Universität Wien* [10]
[7] Linux Infrared Remote Control with Linux as input device: `http://www.lirc.org/`
[8] USB IR Transceiver: `http://iguanaworks.net/`

can be embedded in a stationary or mobile terminal, contains all the loaded communication protocols (cf figure 2) and even specific wheelchair drivers, whereas the client application only has native bluetooth. The client application has been developed using *Qt*, the leading UI framework for devices powered by embedded Linux.

Built on the software architecture presented above, Danah client/server applications still have to be flexible and easily configurable by designers to fit user needs. As for a health care center, deployment at this scale would thus be too complex without high level design tools and techniques.

4. Towards a complete home automation solution

On top of software design (section 3) and hardware implementation (section 2), it is essential to introduce a specification step (cf figure 3) to offer a complete home automation solution, that is flexible and adaptive. Design tools and methods, currently studied and developed in two projects, make it possible to specify both user-system interactions, but also deployment and design of home automation terminals.

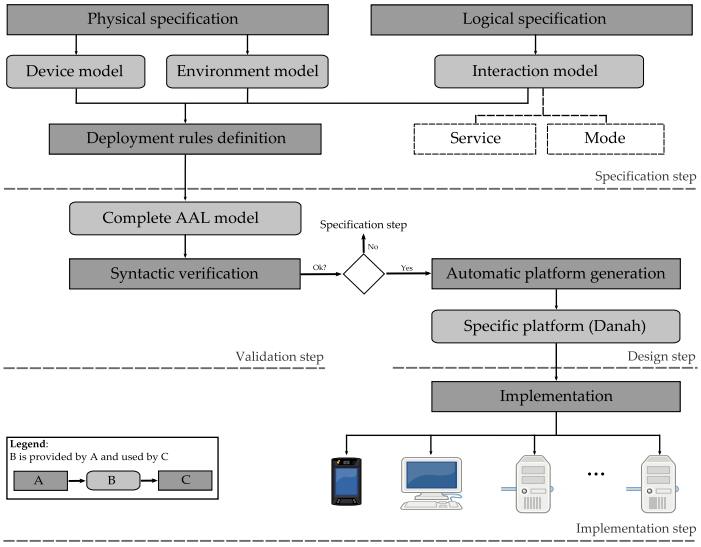

Figure 3: Design flow for ambient assisted living

First of all, services must be tailored to specific user needs. A graphical DSL [9] is currently developed to take into account the specificity of each user. First, the non-expert designer (occupational therapist or relative) describes the environment topology. From devices described in the *environment model*, the designer then defines the *interaction model*, which considers the way the user should be able to interact with his environment. According to user needs, various *services* such as scenario, conditional, or semi-automatic services can be defined. To complement this, *modes* make it possible to put restrictions on the home automation environment. This high level description language, platform independent and expressive enough to fit user needs, is coupled with a model driven flow [12] based on a Model Driven Architecture (MDA). From this platform independent language, designers can automatically generate platform specific code for a home automation application (e.g. Danah described in section 3). MDA is also a flexible and evolutive solution for the integration of new software platforms or hardware targets.

New hardware and deployment specifications also have to be considered. Smart electronic devices can first be integrated in the retained architecture, such as the *Elfel* camera that can embed a processor, a lightweight FPGA [10], and memory storage. Then, it also has to manage the dynamical aspect of the environment and the variety of services which has design implication, in software deployment, reliability and real-time constraints. Computationally expensive algorithms for scenario identification [2] or

[9] Domain Specific Language for home automation dedicated to dependent people

[10] Field Programmable Gate Array designed to be configured by the customer or designer after manufacturing

alert management [13] directly based on the use of home automation and multimedia services require huge calculation and storage capacities. From the latter works, new trends appear and need to be considered to improve design methods for AAL and give new perspectives that have to be explored. The *device model* should make it possible to deploy greedy resource algorithms on several adapted smart devices to complete the calculation, keep real-time constraints, and satisfy QoS. Also, assisted living services should be dependable and fault tolerant. This is why reconfiguration service methods can be implemented and service execution can be shifted onto devices with similar capabilities.

5. Conclusion & Perspectives

An original AAL solution has been presented in this paper. Hardware architecture, based on user, stationary, and mobile terminals, make it possible to offer both environmental control, as well as wheelchair navigation. The Linux execution environment guarantees interoperability between these heterogeneous platforms. The software part then benefits from the inheritance and sharing of the Linux community, including free packages and libraries for communication protocols. This practice allows one to save development time, customize the application and better fit both hardware specification and user needs. Hardware support for GNU/Linux, which has greatly improved in the last decade, has helped enrich the hardware architecture with FPGA or companion robots; such capacities make it possible to integrate an embedded Linux target. Moreover, the introduction of computer-aided design to make home automation solutions adaptable and flexible, leads to new perspectives. First, knowing who is where and what they are doing is central to enabling adapted intelligent behavior, specifically when dedicated to people with disabilities. From the model-driven architecture presented above, ontologies, which are useful to model and reason about context information, can be generated. Then, modeling and characterizing environment resources make it possible to adapt service execution following hardware architecture and to propose reconfiguration service methods.

Bibliography

1. World population ageing 2009 – New York: United Nations Department of Economic and Social Affairs
2. T.B.T. Truong, F. De Lamotte and J.P. Diguet – Proactive remote healthcare based on multimedia and home automation services – Proc. of the IEEE Conference on Automation Science and Engineering (CASE 2009)
3. HomeWorks lighting control – LUTRON – website: `http://www.lutron.com/`
4. Varuna security system – HESTIA – website: `http://www.hestia-france.com/`
5. Multiroom ambient system for video and audio – CORVO Technology – website: `http://www.corvo.com/`
6. Prodigy home automation system – CRESTRON – website: `http://www.crestron.com/`
7. Thebis home automation system – HAGER – website: `http://www.hager.fr/`
8. Linux Home Automation – website: `http://www.linuxha.com/`
9. F. De Lamotte, JP. Departe, F. Le Saout, JP. Diguet and JL. Philippe – Quatra: Final report – Technical report (in french). Lab-STICC, 2008 – see demo here: `http://www.youtube.com/watch?v=T6GCFnkLTc0`
10. W. Kastner et G. Neugschwandtner and M. Kögler – An open approach to EIB/KNX software development – Proc. of the 6th IFAC Conference on Fieldbus Systems and their Applications (FeT '05)
11. S. Lankri, P. Berruet, A. Rossi and J.L. Philippe – Architecture and models of the DANAH assistive system – Proc. of the 3rd Workshop on Services Integration in Pervasive Environments (SIPE 2008)
12. W. Allègre, T. Burger and P. Berruet – Model driven flow for assistive home automation system design – Proc. of the 18th Congress of the International Federation of Automatic Control (IFAC 2011)
13. T.B.T. Truong, F. De Lamotte, J.P. Diguet and F. Saïd-Hocine – Anomaly detection with virtual sensors based on observation of home automation service use – Proc. of the 32th IEEE Conference on Information Technology Applications in Biomedicine (EMBS 2010)