Towards the Deployment of Open Platform AAL Services in Real Life-advantages and Lessons Learned

uSmAAL: A Case Study for Implementing Intelligent AAL Services in Real Life based on the Open Platform universAAL

Jan Stengler, Gouri Gaikward and Helmi Ben Hmida*

Fraunhofer Institute for Computer Graphics Research IGD, Fraunhoferstr. 5, 64283 Darmstadt, Germany

Keywords: Showcase, AAL Services, Standardization, Real Life Deployment, Semantic Interoperability.

Abstract: Nowadays, most Ambient Assisted Living systems are confined to individual projects. They are primarily closed products with a limited set of features, thus reducing its extension, adoption and reuse. The aim of this paper is to make a first attempt to increase standardization and interoperability oriented efforts by focusing on open systems. We aim to share our experience with developing and deploying Ambient Assisted Living solutions on the top level of the standardized open platform universAAL in real life. This paper identifies the essential aspects of the system architecture and investigates the advantages of providing generic services, shared and reusable components in real life. In addition, this paper presents an evaluation protocol of the different components, focusing on system sustainability and reliability.

1 INTRODUCTION

Ambient Assisted Living (AAL) is an umbrella concept for a multitude of devices, services and information and communication technologies (ICT) assisting citizens in their own home and supporting their mobility by means of innovative technology (Wichert and Eberhardt, 2011). Through the literature study, several landscape papers come with the conclusion that despite the huge effort in the field of AAL, the real world exploitation of the platform and research output seems to be fairly limited to a minority of closed systems, specific devices and protocols (Memon et al., 2014). Thus, the real life AAL services use frequency is always under the expectation, where sharing software component, exchanging hardware and easily adding new services component seems to be too far from the application scope (Wichert et al., 2012). As a solution, the best practice for interconnecting different standards, exchanging data between services and easily integrating new services will be the key for development of new generation of AAL services and technology (Kung and Jean-Bart, 2010).

Nowadays, open platforms are supported by a large community of stakeholders, often have no ownership of the code, and usually adopt open standards (Boudreau, 2010). Following this tendency and due to the fact that interoperability, standardization add to security are reported and judged as the major IT challenge in Europe, we believe that it is not efficient to develop AAL solutions for individual project.

Looking forward to face this challenge, we aim to minimize the gap among the real world use case requirements and the capabilities of today's available standard open platform solutions through demonstrating an AAL use case based on the European standard open platform *universAAL* (Tazari et al., 2012) for AAL service implementation and deployment in real life. In the next section we will make an attempt to cover the actual AAL system including its architecture, advantages and the system performance through the evaluation framework.

2 RELATED WORK

AAL technologies are nowadays used for supporting, assisting, preventing and improving wellness and health conditions for different categories of people, mainly the elderly ones (Stevenson, 2014). A smart environment can be presented as a regular looking

*Corresponding Author: Helmi Ben Hmida, Fraunhofer IGD, Fraunhoferstr. 5, 64283 Darmstadt, Germany. E-mail: helmi.ben.hmida@igd.fraunhofer.de.

Stengler J., Gaikward G. and Ben Hmida H..
Towards the Deployment of Open Platform AAL Services in Real Life-advantages and Lessons Learned - uSmAAL: A Case Study for Implementing Intelligent AAL Services in Real Life based on the Open Platform universAAL.
DOI: 10.5220/0005450200670074
In *Proceedings of the 1st International Conference on Information and Communication Technologies for Ageing Well and e-Health* (ICT4AgeingWell-2015), pages 67-74
ISBN: 978-989-758-102-1

home by the installing different types of sensors, actuators and diverse intelligent system where the information from different sensor will help to provide a very good overview about the environment status. In fact, the deduced context will be mainly used for providing, guiding and orienting the system behaviors, and decisions toward safer, healthier and more comfortable environment. Most of AAL systems included in smart environment aims to improve comfort and security, therefore dealing with medical rehabilitation, monitoring mobility and physiological parameters, and delivering remedies (Dlodlo et al., 2013). Thus, many research and development projects are private financed or funded by international and governmental organizations.

Some of the AAL application occupancy includes helping with mobility and automation (Dubowsky et al., 2000) and also increasing the social connection and communicating tools between peers, family and friends (Mynatt et al., 2001). The ACHE system (Mozer, 2008) in Colorado aims to permanently monitor the environment, control the resident action, thus saving energy resources in desire with the resident. The Georgia Aware Homes system is based on intelligent sensors which in return provides information to robots which help to assist, monitor and support elderly people (Kidd et al., 2009). In Japan, almost 15 care apartment have been built, where the objective is to improve the quality of life of both elderly people and their caregivers. From the perspective of medication management and health support related domains, the related AAL systems aim to take control of the heath conditions of elderly people (Qudah et al., 2010), (Khan et al., 2010), where the safety of the elderly people plays a big role in the future of AAL systems. Many Smart home have also been developed in European countries. In UK, the Gloucester's Smart House mainly are geared up to help subjects with dementia by monitoring houses all day (Barnes et al., 2008). In more sophisticated project, the US Hospital on Home project aims to bring hospital to home through implementing a tele care system (Paul III, 2013). Liao et al. have created a robust sleep monitoring system based on the vital sign analysis (Liao and Kuo, 2013).

2.1 Discussion

Although the diversity and the high quality of the available AAL solutions, they have a main common factor which is their "CLOSED" aspect. Closed platform are considered as a software system where only the service developer and/or provider have access and control over the application. From devices interconnection point of view, only pre-implemented sensor and actuator can be taken in consideration, where no new functionalities, sensor and actuators can be straightforwardly added to the originally exploited ones.

2.2 Open Platform

Within the broader question of how to best organize the development and commercialization of the technology as ambient intelligence system, an innovator may choose to "open its technology". Thus, allowing outsider to participate to its development, growing and commercialization. This will help mainly to the improvement of individual component; creation of extension, add-ons and upgrade; the elimination of bugs and error, quality and cost improvement (Tazari et al., 2012). An Open Platform is a software system build based on open standards which implies that the vendor allows, and perhaps supports the ability to collaborate and contribute toward the development of several "compatible" solutions. Thus, by using such a platform, , all hosted open platform AAL services are imperatively interoperable and can interact and survive with each other providing flexibility and freedom to fit the end user's needs.

Tazari et al have presented the AAL open platform universAAL. It has consolidated the work done among other European projects to produce a "Standard" AAL system for large scale deployment of AAL solutions. Similarly, Schmidt et al has started from the assumption that due to the diversity of AAL solutions, the scope of its commercialization is also reduced. Looking at this hypothesis, they only have suggested an open Middleware that can be used to implement, develop and configure AAL solutions. Besides, after a deep framework literature study (Spitalewsky et al., 2013), most of open platform provide a "partial solution, and are not ready to support the full pledged solution ready for real world deployment". As a main consequence based on open platform, we can synthesize that one of the major challenges of the actual AAL domain, is to test, validate and deploy the created platform with all their capability of openness, interoperability, service integration and data sharing in real life scenarios. Or else, what are the main requirements for a successful deployment of an AAL service based on open platform in real environment? How can we judge that open platform is sufficiently mature, sustainable and reliable, thus ready to be installed in real life? To specify the necessary requirement, and to answer this entire question, next sections will make an attempt to bring the universAAL open platform from research to industry. This will be done by presenting a use case of AAL services deployment in real environment and demonstrating the full capacity of open platform through tracing our evaluation framework.

3 SYSTEM ARCHITECTURE

3.1 uSmAAL System Overview

The universAAL Smart AAL System (uSmAAL) forms an open, flexible, reusable and mainly expandable system for providing smart AAL services based on the open source platform universAAL. As seen in Figure 1, uSmAAL inspires its robustness, flexibility and extendibility from its system architecture. It consists mainly of five main blocks: hardware, controller, exporter, AAL smart services and the universAAL open platform. From a Service oriented Architecture (SoA) point of view, the uSmAAL system rely on the semantic open platform uAAL which can totally avoid a priori agreed APIs by purely relying on externalized ontological models so that not only data can be shared based on a universal solution for data representation, but also control functions can be accessed in a goal-based way using one single API. Such an approach can be classified as true semantic interoperability which enables sharing both data and functionality based on pluggable ontologies. Therefore, the uSmAAL will have the ability to:

- Read data from various integrated sensors and combine them to recognize relevant situations.
- Utilize functionality provided by diverse devices distributed in the intelligent space and orchestrate it.
- Achieve an effective behavior in response to the recognized situations.

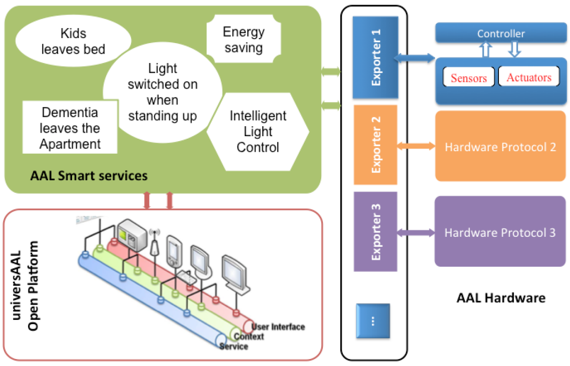

Figure 1: uSmAAL System Overview.

Despite the above cited success key and innovative features, several rumors have unfortunately accompanied platforms born within the research institutes and the public funding. Accompanied reasons are related to "proof of concept" thus doubt about the platform maturity, reliability, stability and robustness to reach the market breakthrough. Several interested industries have always asked the question: "how far is the platform able to resist to long time execution under stressed condition". The suggested system architecture will refute rumors about the research origin software component since the system has proved to be highly stable and reliable while deploying in real life environment.

3.2 uSmAAL Current AAL Services

Adding to the complete overview of the robust and functional system running on the top of an open platform in the next subsections, we will start with the AAL service block as one of the main uSmAAL system components since it presents the "functional" one, where all the interaction with the hardware side and the platform are done through its intermediate. The uSmAAL system has created and installed several AAL services that aim to maintain information like status, events or consumption situation about several electronic sensors and actuators and control them. The services provide access to sensors as well as actuators within each of the homes. Table 1 presents a sample of the implemented AAL services.

Table 1: A sample of uSmAAL AAL services.

AAL Service	Service description
Dim Light	Automatic control of the light system (value in %)
Get Motion status	Detect persons activities
Control Heating	Controls the heater through setting a specific modus)
Get Person position	Detect the person position
Fall detection	Automatic recognize falls
Alarm function	Automatic control of the alarm system (on/off)
Get CO2 value	Get information about the CO2 value in a room

3.3 UniversAAL Platform

The above highlighted uSmAAL individual AAL services are running on the top of the semantic open platform universAAL (Tazari et al., 2012). The latter one has been built in a community effort since 2010 and finally established in January 2014 thus combining the best approaches of EU FP6 and FP7. It provides the result as an open source software platform for AAL domain. The universAAL platform is an open-source software platform for the development, operation and marketing of AAL applications. It is offered with the Apache Software-License 2.0 and is especially designed for the development of open and distributed AAL systems. As a semantic open platform and thanks to its unique features, universAAL will allow the creation of a sustainable ecosystem of AAL applications and service providers. From developer point of view, the open platform-universAAL offers a set of development features that are relevant for development time:

- Assisted development environment to create AAL application based on the universAAL open platform.
- Distribution of AAL services over different processing components in a network
- Sharing information and composing services between all developed application and services upon universAAL.
- Cross-exploitation of the offered functionalities.
- Facilitating the extension of an application in terms of functionalities and devices.
- Simplifying the adaptation of a certain application to new circumstances.

3.4 Exporter

After highlighting the software level, mainly composed of the platform and the uSmAAL AAL services, the exporter aims to build the interface between the real world presented by the hardware and the related protocol from one side, and the virtual part presented by the services and an abstraction of the hardware devices from the other. It receives commands from the AAL services and translates them into low level instruction depending on the addressed controller. This means if a specific device is turned on by some AAL service, the exporter receives this command and redirects it to the corresponding controller. Based on the hardware protocol, it transfers the latter one to the addressed device and vice-versa. If a device is manually turned on, the AAL services have to be instantly notified about this through the information transferred from the device to its controller, then to the service through the exporter. Due to the fact that the exporter builds the bridge between the services and the hardware devices, it also represents a core component of the interoperability in uSmAAL. Due to the platform capability from one side, and to the OSGI framework capacity from another, the exporter has knowledge about the existing controller and their hardware devices. Thus a developer of an AAL service can concentrate on implementing different AAL service without taking in consideration the specific controller and hardware protocol.

3.5 Controller

A controller specifies a standard for the communication where different hardware devices, mainly in a smart environment, are connected to it.

Within the uSmAAL system, the controller aims to establish the communication between the AAL services and the device layers, it can track the state of these devices and perform actions that will change their status. To ensure a maximum capability of interoperability, the uSmAAL system makes it possible to combine different controller in parallel. Examples are PLCs (Programmable Logic Controller), KNX or ZigBee controller. In case of a PLC, the devices will be connected to its modules and controlled through the written value in PLC memory. The controller will receive the commands for controlling devices from the exporter and change the value in the memory of the corresponding module accordingly. For example a lamp can be turned on or off by writing the required value in the memory address of the module the light is connected to. If the status of a device is changed manually, the controller will notice this and the value in the memory address is changed too. In such a case the controller will report this change to the exporter.

3.6 Hardware

The current hardware block of the uSmAAL system is mainly defined by two components: actuators and sensors. The actuators represent hardware devices which can be controlled, e.g. lights, heaters, outlets, blinds, etc. This means it is possible to change the status of the device by performing some interaction. This interaction can be performed by directly changing the actuator values, or automatically through the system itself (e.g. reasoning). In both cases the controller will perform an action which will result in the desired outcome. It may also be possible that an actuator is controlled manually like switching a lamp by pressing a button. If so, the controller will be automatically notified about such an action.

The second component is represented through the sensors. A sensor measures values and registers modifications in the environment and notifies the controller about it. A sensor can be for example a motion sensor, a temperature sensor or a CO_2 sensor. Since sensors are measuring specific variables of the environment (temperature, CO_2 value, etc.), they can keep track of the actual status of it. Thus they build the basis for specified rules and reasoning. For example reasoning like “turn on the light in a room if there is movement at night”, is based on the sensors for movement and daytime.

4 SYSTEM ARCHITECTURE ADVANTAGES

Figure 2 illustrates the flexible modular architecture of uSmAAL. Because of the possibility to use several exporters, the uSmAAL architecture ensures a maximum of interoperability. This means all the hardware components are editable, exchangeable and are able to work in parallel. The AAL services are implemented in the different plug-ins, which are establishing the communication to the exporters. It is possible for a plug-in to interact with more than one exporter and thus control many different types of hardware devices.

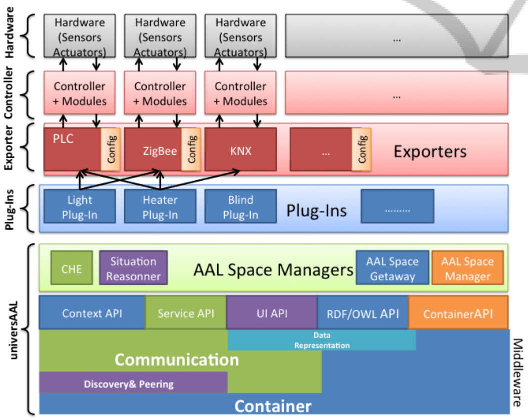

Figure 2: uSmAAL System Overview.

4.1 Interoperability

Interoperability describes the extent to which diverse system are able to work together to achieve a common goal. Thus systems and devices can exchange data and hardware resource easily. Adding to its reliability, stability and auditability, interoperable solutions based on open platform offer a very important cost advantage for their end users. Consequently, better adherence to open sources will permit competition in the market, thus reducing vendor lock and price monopoly. Last but not least, interoperability offers a big room of flexibility and freedom to fit the end user’s needs. uSmAAL system has profit from the universAAL platform capability and enriches it based on its modular architecture and the use of Exporter as an interface between the real environment (Hardware) and the abstract one (Services). In fact, the exporters are defining a protocol for the communication between the plugins and the hardware devices, no matter where they belong to; smart home, eHealth or other linked domain to the AAL.

In fact, cutting the direct link between the AAL service plugins and related hardware offer the possibility for changing and/or adding new hardware technology without being restricted by the solution constraint. This was proved with the uSmAAL system through changing the PLC controller and related device with a KNX one, with the use of the same AAL service.

Adding to it, the designed architecture of uSmAAL offer one of the main needed capability by AAL system which is the system integration, where several new AAL services “plug-ins” can be easily added to the actual ones, then dynamically linked to the appropriate hardware layer.

4.2 Reasoning and Service Evolution

As observed from the beginning, the uSmAAL system as a SoA is classified as a “modular” architecture, since it’s mainly composed of several basic AAL services able to respond the elementary functionalities, dynamically communicating together due to the available semantic layer. The modular approach, as a key for future programming technology, aims to divide the set of complete application to basic modules. It intends to facilitate the re-combination of modules differently, thus offering several new sets of services and/or recognized states dynamically and in a more flexible way.

The recombination process is reinforced through several components developed within the universAAL system, the situation reasoning and the Drools reasoner. Based on these components, it will be too easy to separately create new intelligent rules, thus combining several individual AAL services and coming with new situation or a combination of AAL services. For example, one key technology installed behind the uSmAAL is the localization of residents in their homes and neighborhoods. Once installed, many services can be placed together such as modular fall detection, intrusion detection, alerts when leaving the house or if the window is still open; and much more, Figure 3. Modular technologies will be the main key toward offering better services with minimum cost.

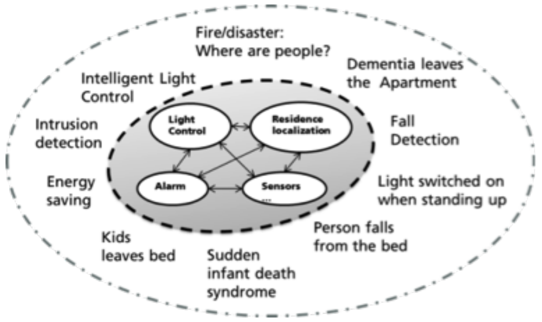

Figure 3: Modular Approach Overview.

5 EVALUATION

Let us recall that the purpose of this paper is to scientifically reject all negative feedbacks around open platforms, mainly the fact that they are not stable, neither reliable and have never been tested in real environment and only used for research. This will be demonstrated via an opposite example: the uSmAAL system. Indeed, the suggested system solution is created upon an open platform, thus profiting from all the advantages previously highlighted in the earlier section. The exploitation of the uSmAAL system will only make sense if it is in parallel tested and proved. It is therefore crucial for the evaluation to include real pilot sites and test in realistic conditions. The evaluation framework should assess a system's quality, which thus will be held to valuable information in addition to hands-on practical information on the performance. To do so, the evaluation framework assesses the uSmAAL system reliability and sustainability. Unfortunately, the literature study revealed that currently no AAL specific evaluation model exists. We aim through this section to evaluate the uSmAAL system such that each part and the whole system functions successfully and with stability.

5.1 Evaluation Approach

The uSmAAL system performance will retain open till finding a concrete answer, mainly related to its sustainability and reliability, to judge its readiness for industrialization and deployment in real life environment. To evaluate the system thoroughly, we will divide it such that each level can be analyzed and tested individually and together to achieve our goal of a stable and reliable system. As a matter of fact, the stability factor will be validated if every bounded input will be held in a bounded output over an optimal time interval. Concerning the reliability factor, a system is reliable if, including all hardware, firmware, and software, will satisfactorily perform the task for which it was designed or intended, for a specified time and in a specified environment. For example, it is very important to have a stable and a reliable system when checking the connection between the PLC controller and exporter, thus to confirm that the socket connection is successfully established every time. To achieve our goal, the system evaluation will be composed of two main evaluation block: The Functional testing and the Stress testing. While functional testing will help to validate the functionality of each individual service, the stress testing will validate the system stability and reliability.

5.2 Functional Test

The uSmAAL system is actually composed by running single services based on four main ontologies: Physical things, devices, furniture and profiles. Functional testing is a quality assurance (QA) process and a type of black box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, where internal program structure is rarely considered. Functional Testing usually describes what the system does. Within the uSmAAL system, the different AAL services are tested, to check if all are executed accurately to get the required output. Some of the examples of test cases that were tested to check functionality include testing whether the universAAL application runs successfully, the configuration file is parsed successfully, also to check if the devices can be controlled without any errors, each device have their respective correct memory addresses. Each use case is tested for all the devices of the device type. For each test case expected output was obtained, e.g. when a light was turned on, it did turn on as expected and thus the functional test is validated.

5.3 Stress Test

Stress testing is a software testing activity that determines the robustness of software by testing beyond the limits of normal operation. Stress tests commonly put a greater emphasis on robustness, availability, and error handling under a heavy load, than on what would be considered correct behavior under normal circumstances (Zhang et al., 2008). Putting uSmAAL under high usage condition will be the way to validate its robustness and readiness for the market.

5.3.1 Looping the Service Requests

The automatic loop for starting the entire uSmAAL service for all registered devices every minute aims to restart almost 50 services every minute. The test aims to validate the communication reliability and stability between the AAL services and the Exporter. After 2 weeks of test, uSmAAL Service execution is achieved without exceptions and delay. Figure 4 shows the graph of delay in milliseconds for each AAL service execution. Observing the x-axis we can see that during more than 3000 service execution within the uSmAAL system, an execution delay of 1020 to 1030ms is obtained where the service call execution time is only 10ms (AAL Smart Service level) where as the hardware part (AAL Hardware level) consumes more than 95% of the execution time . Taking onto account the complexity of the PLC processing and same the network communication, the delay has kept a stable average along the test period with a granularity of $\pm$30 millisecond due to the above mentioned factor.

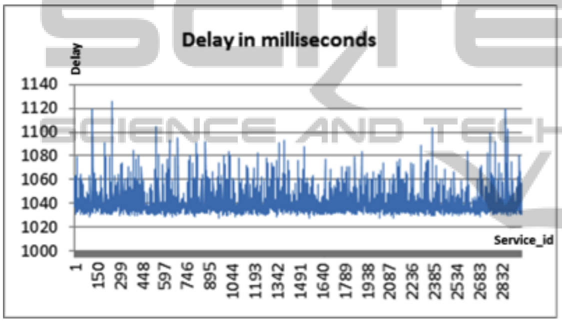

Figure 4: Delay in milliseconds versus service calls.

5.3.2 Proving the Connection Stability Between the Exporter and the PLC

After proving the bridge between the AAL service layer and the Exporter, it is now time to stress the connection between the exporter and the Controller, thus the purpose of this test case is to check the stability of the established socket connection between both layers. After running the protocol for the same test period (2 Weeks), more than 600 000 established socket connections were error free based on the log file.

5.3.3 Analyzing the Service calls

In a more qualitative test, we have analyzed the universAAL open platform log files looking for logged call status for each services. Based on the universAAL platform, the call status can end with values: Succeeded; No_matching_service_found; Response_timed_out ;Service_specific_failure

Figure 5 presents a log portion about the status of a service call. After analyzing all the log files with more than million lines, all services execution are always returned with the ":call_succeeded" status, thus reflecting the reliability of the platform from one side and the uSmAAL system from another one.

```
@prefix : <http://ontology.universAAL.org/uAAL.owl#>
.
_:BN000000 a :ServiceResponse ;
callStatus :call_succeeded ;
:theCallee <urn:uaal_space:8888/7aca5adf-ad0b-
4aa7-a7bf-
471edd80131b#10responder.smp.lighting.server
```

Figure 5: CallStatus succeeded.

5.3.4 Memory Usage

The memory usage issues are considered as most common issues for open platform extracted from the research environment. For the created uSmAAL system, and due to its critical impact on the system future, we have tracked the memory evolution through the standard java tools "visualvm" for more than one month, the memory evolution was always stable and never exceeded a heap of 250MB. Figure 6 presents a sample of the memory evolution.

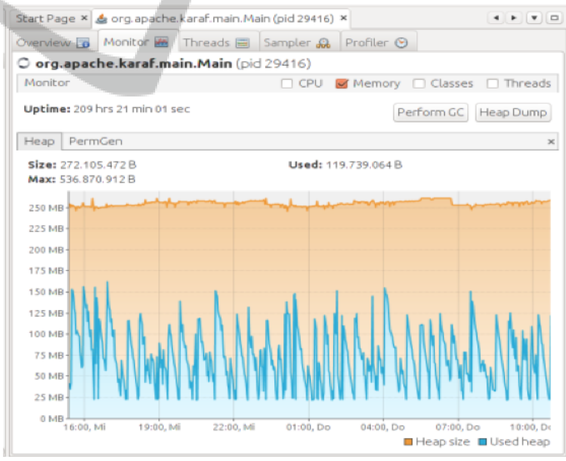

Figure 6: Memory consumption Graph.

6 CONCLUSIONS

Nowadays, existing AAL systems are bound to already implemented services and special types of hardware devices. This lack of dynamic makes them rather unusable for the long term usage in a real life environment. From the other side, and despite the wide spread support for openness and heterogeneity, most projects have not yet sufficiently reached a mature level allowing them to access the industrial market where only a very few AAL open platform architecture are linked to the industry by producing a stable and sustainable system mature enough to break through the market. Moreover, there is no best practice about how to use them in real life deployment.

We have made an attempt through the current paper to refute all negative feedback and misunderstandings around AAL Open Platform. The presented uSmAAL system builds upon the open platform universAAL and has proven a very advanced degree of stability and reliability, while being deployed in a real life environment. It has been recently deployed in 22 apartments for elderly people thus supporting them for their daily activities, and increasing their comfort and security in the meantime. In addition, this paper has presented the best practice related to the combination of the needed components, while deploying it in real environment. The delivered system is expandable. Thus extra-AAL services can be implemented and the set of hardware devices/protocols can be extended and/or exchanged and the set of supported devices can be extended. At the end the required result is obtained and our goal of a stable and reliable system is achieved and approved.

Further future work will mainly articulate complex situation recognition in an AAL smart environment add to the enrichment of the actual service catalogue with more health oriented AAL services.

REFERENCES

Barnes, N., Edwards, N., Rose, D., and Garner, P. (2008). Lifestyle monitoring-technology for supported independence. *Computing & Control Engineering Journal*, 9(4):169–174.

Boudreau, K. (2010). Open platform strategies and innovation: Granting access vs. devolving control. *Management Science*, 56(10):1849–1872.

Dlodlo, N., Smith, A., Montsi, L., and Kruger, C. (2013). Towards a demand-side smart domestic electrical energy management system. In *IST-Africa Conference and Exhibition (IST-Africa), 2013*, pages 1–12. IEEE.

Dubowsky, S., Genot, F., Godding, S., Kozono, H., Skwersky, A., Yu, H., and Yu, L. S. (2000). Pamm-a robotic aid to the elderly for mobility assistance and monitoring: a helping-hand for the elderly. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 570–576. IEEE.

Khan, D. U., Siek, K. A., Meyers, J., Haverhals, L. M., Cali, S., and Ross, S. E. (2010). Designing a personal health application for older adults to manage medications. In *Proceedings of the 1st ACM International Health Informatics Symposium*, pages 849–858. ACM.

Kidd, C. D., Orr, R., Abowd, G. D., Atkeson, C. G., Essa, I. A., MacIntyre, B., Mynatt, E., Starner, T. E., and Newstetter, W. (2009). The aware home: A living laboratory for ubiquitous computing research. In *Cooperative buildings. Integrating information, organizations, and architecture*, pages 191–198. Springer.

Kung, A. and Jean-Bart, B. (2010). Making aal platforms a reality. In *Ambient Intelligence*, pages 187–196. Springer.

Liao, W.-H. and Kuo, J.-H. (2013). Sleep monitoring system in real bedroom environment using texture-based background modeling approaches. *Journal of Ambient Intelligence and Humanized Computing*, 4(1):57–66.

Memon, M., Wagner, S. R., Pedersen, C. F., Beevi, F. H. A., and Hansen, F. O. (2014). Ambient assisted living healthcare frameworks, platforms, standards, and quality attributes. *Sensors*, 14(3):4312–4341.

Mozer, M. C. (2008). The neural network house: An environment hat adapts to its inhabitants. In *Proc. AAAI Spring Symp. Intelligent Environments*, pages 110–114.

Mynatt, E. D., Rowan, J., Craighill, S., and Jacobs, A. (2001). Digital family portraits: supporting peace of mind for extended family members. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 333–340. ACM.

Paul III, D. P. (2013). An innovation in healthcare delivery: Hospital at home. *Journal of Management Policy and Practice*, 14(6):73–91.

Qudah, I., Leijdekkers, P., and Gay, V. (2010). Using mobile phones to improve medication compliance and awareness for cardiac patients. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, page 36. ACM.

Spitalewsky, K., Rochon, J., Ganzinger, M., Knaup, P., et al. (2013). Potential and requirements of it for ambient assisted living technologies. *Methods Inf Med*, 52(3):231–238.

Stevenson, S. (2014). 6 ways ambient assisted living improves quality of life. http://www.aplaceformom.com/blog/10-29-14-ambient-assisted-living/. Accessed: 2014-12-14.

Tazari, M.-R., Furfari, F., Fides-Valero, Á., Hanke, S., Höftberger, O., Kehagias, D., Mosmondor, M., Wichert, R., and Wolf, P. (2012). The universaal reference model for aal. *Handbook of Ambient Assisted Living*, 11:610–625.

Wichert, R. and Eberhardt, B. (2011). *Ambient assisted living*. Springer.

Wichert, R., Furfari, F., Kung, A., and Tazari, M. R. (2012). How to overcome the market entrance barrier and achieve the market breakthrough in aal. In *Ambient Assisted Living*, pages 349–358. Springer.

Zhang, B.-j., Pan, X.-z., Wang, J.-b., et al. (2008). A recoverable stress testing algorithm for compression and encryption cards. *Journal of Zhejiang University SCIENCE A*, 9(10):1398–1405.