Exploiting the universAAL Platform for the Design and Development of a Physical Activity Monitoring Application

Vassilis Kilintzis, Ioannis Moulos, Vassilis Koutkias, Nicos Maglaveras
Institute of Applied Biosciences, Centre for Research and Technology Hellas (CERTH)
6th Klm. Charilaou - Thermi Road, P.O. BOX 60361, GR - 57001
Thermi, Thessaloniki, GREECE
{billyk, moulos, vkoutkias, nicmag}@certh.gr

ABSTRACT
While various technical approaches for constructing Ambient Assisted Living (AAL) applications/services have been proposed, it has become apparent that interoperability of AAL systems is the key challenge that has to be tackled for exploiting AAL technologies in their full potential. Equally important, there is a lack of tools that can support AAL system/application developers to implement systemic and affordable solutions. The objective of this work is to illustrate the design and development of a basic AAL application devoted to physical activity monitoring by exploiting the universAAL open platform and tools. Our main goal is to illustrate the procedure that developers have to follow for such a development as well as the benefits offered from the adoption of universAAL in terms of extensibility and interoperability.

Categories and Subject Descriptors
J.3 [**Life and Medical Sciences**]: *Medical information systems*. K.4.2 [**Computers and Society**]: Social Issues – *Assistive technologies for persons with disabilities*. H.3.4 [**Information Storage and Retrieval**]: Systems and software – *Distributed systems, User profiles and alert services*.

General Terms
Measurement, Design, Experimentation.

Keywords
Physical activity monitoring, biomedical sensors, Ambient Assisted Living (AAL), AAL middleware, ontologies, interoperability, development platform.

1. INTRODUCTION
The ageing population is growing at a considerably fast rate. In Europe, for example, in absolute terms the number of elderly has tripled over the last 50 years and will more than triple again over the next 50-year period [1]. In this respect, Information and Communication Technologies (ICT) solutions and Ambient Assisted Living (AAL) in particular have been proposed as a major pillar for accomplishing the European-wide vision of active and healthy ageing (AHA) [2]. Especially for AAL, an important market potential has been identified [3]. Nevertheless, the AAL adoption is quite limited with respect to the raised expectations, mainly due to the following barriers: (a) significant resources for the implementation of AAL services/applications are required, and (b) major shortcomings towards achieving interoperability among AAL services/systems.

Aiming to overcome the above issues the EU-funded project universAAL (Universal Open platform and reference architecture for Ambient Assisted Living - http://universaal.org/) is conducting research towards the development of an open AAL development platform. universAAL aspires to establish a standardized design and development approach that will make the implementation of AAL solutions technically feasible and economically viable [4].

The objective of this work is to present the procedure of exploiting the universAAL platform for the design and development of a basic AAL application for physical activity monitoring. Through this case study, our aim is to demonstrate the overall approach that has to be followed, and ultimately, the benefits offered by adopting the universAAL platform for AAL application development. Although the application is rather simple and various works targeting this domain have been presented [5], it has been selected as an indicative case study to illustrate the adoption of universAAL for such an implementation.

2. THE universAAL PLATFORM
universAAL constitutes a software platform aiming to: (a) facilitate the development of "assistive systems" by connecting various, heterogeneous devices to a single, unified network and, (b) deliver the means to control this distributed system and its devices. In the scope of universAAL, such assistive systems resemble "intelligent agents" [6], capable of perceiving the state of their environment (using sensors), assessing the perceived state (if it is desirable or not) and, if the state is not desirable, trying to find a way of how to influence their environment (through their actuators) to reach a more desirable state.

In universAAL, a single device that is connected to an assistive system is referred to as being a "node". In principle, there are two ways of integrating a networked device into an universAAL-based assistive system, i.e. by: (a) Installing a part of the universAAL software on the device, the so-called "Middleware", that contains the communication infrastructure of the universAAL platform; all devices that run the Middleware can actively participate in the communication of the system. (b) Connecting the device to a node

PETRA '13, May 29 - 31 2013, Island of Rhodes, Greece

http://dx.doi.org/10.1145/2504335.2504351

that runs the Middleware, and using this node as a communication intermediate. For many devices, e.g., low-power wireless sensors, the latter is the only way of connecting them to the system, as they cannot run additional software besides their firmware.

Central to universAAL is the notion of the “AAL Space” [4]. This corresponds to a smart environment centred on its users in which a set of embedded networked artefacts, both hardware and software, collectively realize the paradigm of Ambient Intelligence (AmI), mainly by providing for context-awareness and personalization, adaptive reactivity, and anticipatory pro-activity. Smart homes and cars are examples of AAL Spaces. Besides the Middleware, the universAAL platform is made up of various other “higher level” software components. Still, the Middleware is the basis for all these components, meaning that these cannot be executed, if the Middleware is not available. The Middleware hides the distribution and heterogeneity of the diverse devices that make-up the core system. Three interaction channels called “Buses” constitute the heart of the universAAL platform, namely, the *Context Bus*, the *Service Bus* and the *User Interface (UI) Bus*. All communications between applications take place only in a “round-about” way via one of the buses, even if the applications are located on the same node. Each of the buses handles a specific type of message/request, and the way that a bus operates is based on its characteristics.

In more detail, the *Context Bus* is an “event-based” channel that is used for publishing information about the state of the environment and/or the AAL system. This kind of “contextual” information is used by the AAL system to recognize situations in which it is supposed to act. The *Service Bus* is “call-based”. An application that offers a service (i.e., a functionality available through the platform) announces this by registering a corresponding “service profile” (i.e., a description of what it is capable of offering) with the Service Bus. It is up to the Service Bus to find matching service profiles to the service request and, if a match is available, to forward the request to the corresponding service callee(s). Once the service callee has completed the requested task, it places the “service response” on the Service Bus, which in turn makes sure that the answer reaches the service caller. Finally, the purpose of the interaction *UI Bus* is to deliver messages that are related explicitly with the users (also “event-based”).

Other important components of the platform include the *Service Orchestrator*, the *Situation Reasoner*, the *Dialog Manager*, etc. In addition to the core software, the platform provides tools to support both developers and end-users to install, use, maintain, and evolve their universAAL-based systems/applications. In particular, developers may use the AAL Studio which includes tools such as the *Ontology Project Wizard*, the *Coding Assistant*, the *uStore Publishing Client*, etc. Formal/informal caregivers and the elderly themselves may browse the available applications and services via the *uStore*. A thorough description of the universAAL platform is available in [4], while full documentation and source code are offered through the universAAL development server (http://forge.universaal.org/gf/) upon free registration.

3. PHYSICAL ACTIVITY MONITORING APPLICATION: A CASE STUDY

3.1 Design

The design and development of an application using the universAAL framework requires conformance with a number of principles. In summary, the procedure that a developer has to follow to develop a universAAL-based application using its Middleware and tools involves the steps listed below:

- *Definition of the necessary application ontologies*: The universAAL platform adopts an ontology-based approach, in order to facilitate interoperability among different distributed systems. Thus, each concept used by an application must be part of a defined ontology. In universAAL, ontologies are modelled as Java classes, so that they can be integrated in the platform.
- *Definition of the Buses to be employed*: This is based on the functionality that the application shall provide. Hence, developers need to comprehend the notion and use of Buses and how the data of an application are transmitted to/from them.
- *Implementation of the required Java classes, application logic and integration with Bus functions*: Overall, the process of developing universAAL-based applications includes the implementation of OSGi (Open Services Gateway initiative - http://www.osgi.org/) compliant Java archives called “bundles”.

External devices (e.g., sensors) may communicate with the universAAL platform by translating their functions in specific forms such as service requests, context events, UI responses, etc. In our case, a simple Physical Activity Monitoring application has been implemented based on a wearable multi-sensing device. Specifically, BioHarness™ was adopted, a commercial, unobtrusive device that monitors various physiological data, such as heart rate, activity level, posture, etc., appropriate for our case (http://www.zephyr-technology.com/products/bioharness-3/). A device-specific communication module was developed, in order to acquire the sensed data.

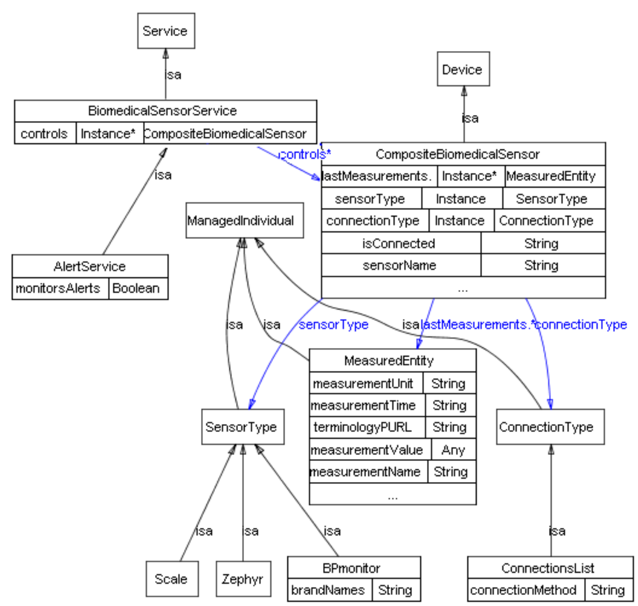

Figure 1. Main concepts of the PhysicalActivityMonitoring ontology.

Following the design principles and steps mentioned above, for the particular application the *PhysicalActivityMonitoring* ontology has been created, aiming to represent the relevant concepts, i.e., sensors, measurements, services, etc. According to the rationale of universAAL, the purpose is to represent the data transferred via the Middleware as instances of the *Resource* class or of its subclasses. The *Resource* class is the universAAL’s representation of an RDF (Resource Description Framework) resource and the top-level class in the ontology framework hierarchy. Each class defined in a universAAL ontology must be a subclass of the *Resource* class.

The respective OWL (Web Ontology Language) ontology that we developed using Protégé (http://protege.stanford.edu/) was transformed to a set of Java classes, in order for the application to use the Middleware. In particular, a Java project including the concepts of the *PhysicalActivityMonitoring* ontology has been developed as a standalone bundle. The main classes of the ontology are depicted in Figure 1. The *CompositeBiomedicalSensor* class (extends universAAL's *Device* class) corresponds to a biomedical sensor containing all its basic functionality and features. Each instance of *CompositeBiomedicalSensor* has a unique sensor type and a connection type. The *SensorType* class describes different kinds of sensors, which can be adopted such as a weight scale, a blood pressure monitor, etc. The *ConnectionType* class is used to define the type of available connections (e.g., WiFi, Bluetooth®, cable, etc.). The sensor type is linked with a set of measurements, expressed as an array of instances of the *MeasuredEntity* class. Instances of the *MeasuredEntity* class correspond to the relevant measurements. Each *MeasurementEntity* instance is defined by the measurement name, value, unit and a Persistent Uniform Resource Locator (PURL) code obtained from an external terminology. The *BiomedicalSensorService* service is responsible for managing the biomedical sensors of the application.

In order to support transmission of the sensed data and their management via the universAAL's Middleware, a client-server model has been adopted. The *Server* has been developed as a standalone bundle capable of communicating with BioHarness™ via Bluetooth®, acquiring the measurements and managing its functionalities (start/stop/get sensor's information, etc.). Respectively, the *Client*, also implemented as a bundle, offers the application's features and the remote management of each sensor via the UI Bus to the potential user. More specifically, the Server is responsible for communicating with a Reasoner service implemented via the Drools (http://www.jboss.org/drools/) rule engine, which is currently deployed as a platform service. The Reasoner uses "consequences", i.e., small parts that encapsulate the information of a fired rule, which are used in order to generate appropriate alerts for the user. This is performed after publishing the context events to the Client, in order to generate the appropriate alert UI Dialogs. Additionally, the use of the Service Bus is required, so that the Client can exploit the functionalities defined and provided by the Server as requested by the user.

From a developer's viewpoint, the *Server bundle* contains Java classes for: (a) creating the required service profiles that will be offered to the service callers through the service callees; (b) receiving service calls; (c) handling messages that are context events; (d) the discovery and peering of a Bluetooth® sensor; (e) receiving and providing measurements-data and handling sensor's status using the communication module, and (f) creating a "list" with all the available (registered) sensors. Similarly, the *Client bundle* contains the necessary Java classes for: (a) creating the required service calls to the Server and getting the appropriate service responses; (b) initiating the main dialog of the application; (c) creating the UI of the different kinds of alerts; (d) generating sound alerts, when necessary; (e) creating the graphical representation of the last measurements and the respective alerts, and (f) containing the user ID or name of the potential assisted person as well as the Client's initialization functions.

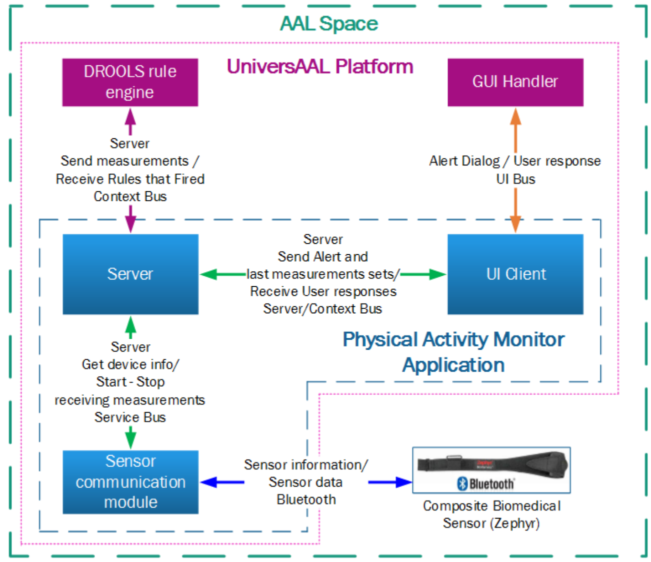

Figure 2. Overview of application modules and functionality.

3.2 Application Scenario

In this subsection we illustrate the exploitation and use of the physical activity monitoring application via an application scenario. The AAL Space for this application can be instantiated for use in various scenarios. These scenarios include either using it as: (a) a module of more complex services, or (b) a stand-alone application. In the first case, an example service could be monitoring the end-user and facilitating the interaction with caregivers outside the AAL Space and, possibly, providing storage of the acquired data to an Electronic Medical Record (EMR), allowing for their later analysis by an expert. In the latter case, the application may be used to: (a) motivate the user during physical exercise and warn him/her in case of excessive exercising or (b) inform caregivers if a handicapped or elder person falls and is not able to recover. Here, the perspective of the stand-alone application will be described from a technical viewpoint.

Let us assume that an elderly person is living in a house where the universAAL platform is installed in several nodes (e.g., a desktop computer, a portable computer / tablet, the TVs placed in various rooms of the house, the mobile phones of the habitants) formulating an AAL Space. The portable computer is situated in the gym room where the user performs his cardio exercises by wearing his monitoring device and using the physical activity monitoring application for safety. Prior to starting his exercise, the user launches the physical activity monitoring application in his node, and selects to use during the exercise the specific device from the various sensors presented as a list of registered sensors/devices in his AAL Space. After selecting the proper device from the list, the user initiates the monitoring session. At this point the user's selection is handled by the Server part of the application, which initiates the Bluetooth® connection. If the sensor responds to the request, the system is ready to start requesting measurements. The procedure of requesting and receiving the response from the sensor via Bluetooth® is handled by the sensor communication module that is responsible for the communication of the selected type of sensor and it is periodically executed (e.g., every 5 sec), while the monitoring session is active. During the monitoring, the measurements populate the *PhysicalActivityMonitoring* ontology. The system is using the universAAL Context Bus (Figure 2), in order to publish the

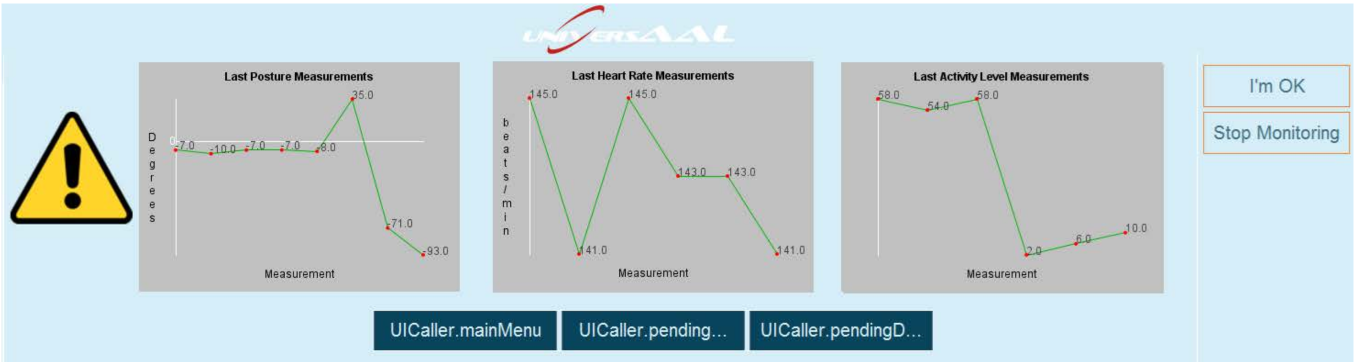

Figure 3. Example of "orange" alert including plots of last measurements of posture, heart rate and activity (please note that the depicted interface is built using the currently available UI components offered by the UI Bus).

measurements. The Reasoner located in the central node of the house is a subscriber for these events, in order to receive and assess the obtained measurements.

At some stage of the exercise session, the user begins intensive cardio work out. The system implicitly monitors his biosignals (heart rate, breathing rate, activity and posture) and publishes them on the Context Bus where applications (like the physical activity monitoring) that have subscribed to the specific context event can use them. As the session progresses, the heart rate of the user is raising significantly. After 25 sec of consecutive heart rate measurements exceeding 110 beats/min, one of the rules defined for the user fires and a unique context event is published, informing the subscribers (i.e., the physical activity application server) that the Reasoner service generated a heart rate alert.

The physical activity monitoring application manages the alerts generated by the Reasoner taking into account the reaction of the user. When the server receives the context event, it forwards it to the UI Client module, which in turn uses the UI Bus to inform the users who are associated with the UI dialogs of the application (Figure 2). The UI Bus is designed to show the dialog in the node that it is closest to the user, so it presents the alert dialog shown in Figure 3 that informs the user for the type of alert along with the last six sets of measurements in line plots. Initially, it presents an alert dialog described as "orange" (i.e., mild type of) alert, indicated by the orange exclamation mark icon, and options to either stop the monitoring or acknowledge the alert. If the user does not react in 20 sec to the prompt and the heart rate measurements continue to cause the rule fire, the physical activity monitoring server resends an alert dialog through the UI Bus. The new dialog is described as a "red" (serious) alert and it is followed by a siren sound to draw the attention of the user or other associated users. By acknowledging an alert dialog, the system resets the state of the alert and continues the monitoring normally.

4. CONCLUSION

The current paper presented a simple, though important AAL application that may support independent living and active and healthy aging. Our aim was to illustrate the procedure we followed for implementing this application by using the universAAL platform and tools, in a rather simplified way, so that developers may comprehend the overall procedure for reproducing the approach for their own implementations. Thanks to the adoption of the universAAL open platform and its scalable and interoperable design, implementation of services/applications like the one presented can be expanded to support services provided by other AAL Space nodes. Our future work includes the development and testing of more complex AAL applications, e.g., using multiple sensors, more advanced functionalities and multiple users, exploiting the full potential and the tools offered by the universAAL platform. Such advancements will be facilitated by the enrichment of the universAAL platform with additional features and components.

5. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n◦ 247950.

6. REFERENCES

[1] The 2012 Ageing Report: Economic and budgetary projections for the 27 EU Member States (2010-2060). 2012. *European Economy*. 2 (May 2012). Available at: http://ec.europa.eu/economy_finance/publications/european_economy/2012/pdf/ee-2012-2_en.pdf.

[2] Steering Group of the European Innovation Partnership on Active and Healthy Ageing. 2011. Strategic Implementation Plan for the European Innovation Partnership on Active and Healthy Ageing. Available at: http://ec.europa.eu/research/innovation-union/pdf/active-healthy-ageing/steering-group/implementation_plan.pdf#view=fit&pagemode=none

[3] Meyer, I. Müller, S., and Kubitschke L. 2012. AAL Markets – Knowing Them, Reaching Them. Evidence from European Research. In *Handbook of Ambient Assisted Living*. J.C. Augusto et al., Eds. IOS Press, 346-368. DOI=http://dx.doi.org/10.3233/978-1-60750-837-3-346.

[4] Tazari, M. R., et al. 2012. The universAAL Reference Model for AAL. In *Handbook of Ambient Assisted Living*. J.C. Augusto et al., Eds. IOS Press, 610-625. DOI=http://dx.doi.org/10.3233/978-1-60750-837-3-610.

[5] Freedson, P.S., Miller, K. 2000. Objective monitoring of physical activity using motion sensors and heart rate. *Research Quarterly for Exercise and Sport*. 71(2 Suppl):S21-9.

[6] Wooldridge, M., and Jennings, N. R. 1995. Intelligent agents: theory and practice. *The Knowledge Engineering Review*. 10 (2), 115-152. DOI=http://dx.doi.org/10.1017/S0269888900008122