Flexible Management of Ambient Assisted Living Environments by Ontology-Based Configuration

Tom Zentek, Jürgen Bock, Asarnusch Rashid

FZI Forschungszentrum Informatik
Karlsruhe, Germany
Email: [zentek|bock|rashid]@fzi.de

***Abstract*—Ambient Assisted Living (AAL) environments are Smart Environments supporting elderly people and their caregivers in their daily life. Such environments are characterised by a wide heterogeneity of software, hardware, and human components as well as rapidly changing needs of the assisted persons. Therefore, flexible management is a crucial success criterion for AAL environments that has not been addressed in the state of the art of AAL research yet. In this paper, we propose an ontology-based configuration framework that fills this gap. Using ontologies to describe AAL environments enables a common understanding and a reusable semantic representation of AAL use cases. This improves the flexibility needed to configure the environment in order to deal with changing individual needs of the assisted persons.**

***Keywords*–ambient assisted living; configuration; semantic; knowledge-based configuration; OWL; smart environment; ambient intelligence**

I. Introduction

With the increasing development of Internet of Things technology, the vision of Smart Environments becomes real. Smart Environments consist of hardware (e.g. sensors) and software services linked together and embedded "in everyday settings and commonplace tasks" [1]. Ambient Assisted Living (AAL) is an application of Smart Environments, dealing with the assistance of elderly peoples and their caregivers in their daily life. AAL hardware is installed in the environment of the assisted persons. An *AAL platform* collects the data from the connected hardware and provides software services to process the collected data. Also, it controls the hardware and interacts with the assisted persons. The different *components* from the AAL environment e.g. software, hardware and human resources can be grouped into *AAL use cases*. End users of these systems are *assisted persons* and their caregivers. *Deployers* provide end user support in installation and maintenance of the AAL environment. *Developers* are responsible for the AAL software and AAL hardware.

Regarding the management of an AAL environment, two factors are mainly challenging:

Heterogeneity of AAL solutions: In everyday settings end users will use AAL components from different developers in their home environment, but they want to use the same hardware resources and share data among different software services. So far, manufacturers and technology providers develop their own standalone solutions, each of them having their own unique data representation.

Individual and changing needs of the users: Users have individual settings (e.g. home environment, familiar circumstances) and individual demand for assistance (e.g. mental state, physical condition). Furthermore, the elderly have rapidly changing needs, depending on their current state of health and general condition.

These two challenges are addressed by the flexibility of an AAL environment as a "measure of how easily a system's capabilities can be modified in response to external change" [2]. Therefore, we conclude that flexibility is a crucial success criterion for the management of AAL environments.

Previous work does not address this gap yet: Due to the consolidation and enhancement of a variety of AAL frameworks that had been developed in research projects over the last years [3], [4], [5], [6], one can suggest that today semantic middlewares have established themselves. They use formal languages such as Resource Description Framework (RDF) , Web Ontology Language (OWL) and Business Process Execution Language (BPEL) to gather declarative knowledge, service requests as well as procedural knowledge in a machine readable form and to realise use cases. So far, this formal description has been a very suitable but static basis for configuration aspects. In literature, one can find approaches for ontology-based configurations in other domains. Ardito et al. [7] describe how orthogonal data sources can be connected for configuration purposes by means of ontologies and mapping processes. Dong et al. [8] attempt to simplify configurations with the help of reasoning based on a semantic description. However, none of these approaches in their present form are directly applicable to management of an AAL environment and the above described challenges. Further preliminary work can be found in the field of smart homes. For instance, there are publications [9], [10] about the configuration and personalization of intelligent houses adjusted to the residents requirements. However, semantic approaches and a changing medical context in terms of AAL have not been addressed so far.

Ontology-based configuration as a specific implementation of knowledge-based configuration [11] provides a possible solution to fill this gap. The two core advantages of an ontology [12] used in this approach are: (i) Reusable ontology concepts can ensure the interoperability of the different use case components. (ii) They provide a common understanding for the knowledge representation in the environment and its stakeholders.

Therefore, this paper investigates the application of an

 ISBN: 978-1-61208-421-3

ontology-based configuration to increase flexibility of the management of AAL environments and contributes to the state of the art by presenting an ontology-based configuration framework. This approach allows for flexibility in the phases [13] of development, design, installation, customisation and maintenance of the implemented use cases including their components: software, hardware and human resources. Furthermore, some of these phases can be automated and all of them can be tool-supported.

As an example, think of an elderly person getting up from bed at night to have a glass of water. The use case "fall-prevention" uses a controlling software component processing data from motion detection sensors and actuators to light the way to the kitchen. In our approach the light source would be a general concept reusable for managing all lights on the way to the kitchen or even in the house. The flexibility gained by our approach can be shown in this example by considering that different motion sensors and light switches from different hardware manufacturers are installed in the environment. With increasing dementia of the resident it is necessary to integrate a second use case. It allows the monitoring of the usage of the front door to ensure his or her safety. Therefore, the information about his current location in the environment is needed and a sensor to detect the door state. The already installed motion detection sensors can now be reused for the new use case and a contact sensor has to be mounted at the front door. Now two services ("fall-prevention" and "front door control") are interoperable using the same hardware based on the same reusable ontology concepts.

This paper is structured as follows. Section II describes the requirements for the AAL environment management, collected by a stakeholder analysis. Based on this input, Section III demonstrates the developed configuration framework with the according middleware, ontologies, processes and tools. Subsequently, Section IV presents the implementation of the framework. A discussion of the results concludes this paper in Section V.

II. Configuration Requirements in AAL Environments

The requirement analysis is divided into a stakeholder analysis and subsequently into expert workshops. For the stakeholder analysis, results of the standardisation group AALIANCE [14] are used to identify the main stakeholders in the management of an AAL environment. Second, a workshop with domain experts (representing the identified stakeholders, 10 participants) was organized to gather their configuration requirements. In a structured discussion guided by the deployment process from five other AAL platforms (Persona, Soprano, MPOWER, OASIS, AMIGO) the experts finally came up with a set of stakeholders and requirements.

Five different groups of stakeholders [14] are involved directly into the phases (development, design, installation, customisation, maintenance) of the configuration of an AAL environment. The following list gives an overview about the defined subset of stakeholders which are directly involved in the configuration. The group of developers is split in two main roles: 1. The *software developer* implements the application; 2. The *hardware developer* manufactures the devices used by the software. After purchasing a software, hardware or acquiring human resources components, the group of deployers ensures the integration of the components in the AAL environment. The majority of this group (assisted persons) is not directly faced with the configuration process, but has to be involved as they need to provide information about themselves and want the AAL environments to be configured for their needs.

Altogether the expert group came up with 63 technical requirements [15]. They can be summarised to the following four main requirements:

- R1. Easy Integration of new functionality: A flexible management shall allow for easy integration of new software, hardware, and human resources.
- R2. Shared Collection and Provision of Information: A flexible management shall enable the stakeholders to share information about the AAL environment between each other in a common way.
- R3. Adjustment to End User: A flexible management shall allow for a comfortable way to adapt different software, hardware and human resources to a wide variety of different end user and their needs.
- R4. Maintenance of the AAL System: It should be possible to maintain the installed software, hardware and human components easily.

III. Configuration Framework

To address the above mentioned four requirements R1, R2, R3 and R4 a configuration framework was developed based on a semantic middleware universAAL [16]. It is implementing the AAL platform of the AAL environment. With such a middleware it is possible to use ontologies within different use case components. The ontologies allow for reusing knowledge about the AAL environment implemented by another use case component. Together with the common understanding between components, this is the base to make heterogeneous components interoperable.

For the organisation of the AAL environment management a configuration process is developed. It defines the phases influencing the configuration of an AAL use case beginning with its development and ending with its maintenance. Tools are developed allowing for editing special parts of the ontologies and to assist the stakeholders through the configuration to provide an easy and powerful AAL environment management. The development of the ontologies, the configuration process and the tools followed an iterative approach. In the first iteration, six participants in a focus group defined the first draft of the ontologies and the configuration process. For a proof of concept at the end of this iteration, a first implementation of the configuration framework (named "Jambi") was developed and installed in five living labs)CIAMI Living Lab TSB (Spain), CERTH-HIT und CERTH-IBBR Labs (Greece), Laboratory of Environment Intelligence UPM (Spain), Smart Home AIT (Austria), FZI Living Lab AAL (Germany)) [17], [18]. These experiences were used in the second iteration to revise the ontologies and the processes with seven participants in a focus group and to implement a second prototype (named "Vaadin"). Additionally, the second step involves creating tool support for the stakeholders to ease the usage of the configuration framework. Finally, the second prototype is currently tested in nine real homes (Valencia (n=5), Madrid (n=3), Vienna (n=1)).

 ISBN: 978-1-61208-421-3

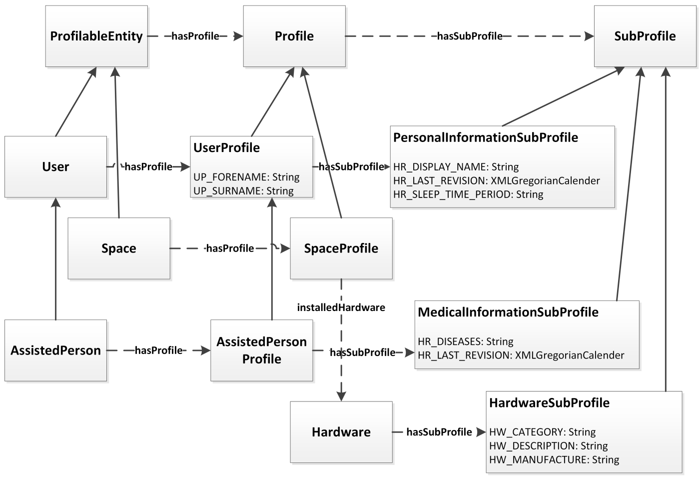

Figure 1. Configuration related ontologies.

A. *Profile Ontology*

To ensure the interoperability between use cases the information about its components is represented in ontologies. They are delivered with the use case component and integrated in the AAL environment with the configuration framework. In order to extend the middleware ontology concepts with a new ontology, coming for software, hardware or human resource components, one has to follow design patterns. On the highest layer the middleware ontology provides three concepts (cf. Chapter 1): Every use case component (software, hardware or human resource) extends the `ProfilableEntity` in order to create new sub-concepts for its own usage. Every `ProfilableEntity` (Domain) can have one `Profile` (Range). Their instances are connected via `hasProfile` (Property) to represent general Information about the software, hardware or human resource components. Every `Profile` (Domain) can have several `SubProfiles` (Range). Their instances are connected by `hasSubProfile` (Property) to represent special information needed by the components for the use case purposes.

For the example use case “fall-prevention” the ontology has to be extended with concepts in the following way (cf. Figure 1). For the human resource part a new concept of an assisted person is needed which automatically inherits the `PersonalInformationSubProfile`. This sub-profile implements a data property to define the sleeping time period (time in which the lights have to be switched on at night). All used functionality of the motion detection sensors and light switch actuators is modelled in the `HardwareSubProfile`. For new concepts, for example a picture of the hardware, a new data property linking to the picture or even a whole new `SubProfile` would be created (e.g. related to the manufacture (`HardwareManufactureSubProfile`) or the communication protocol). The concept of spaces is used here to group hardware related to different locations.

Furthermore, for the second use case with the increasing dementia a new sub profile (e.g. `MedicalInfomationSubProfile`) can be integrated and reuse previously defined ontology parts.

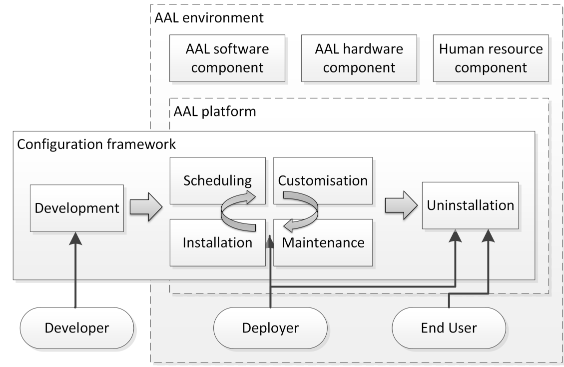

Figure 2. The configuration framework with its stakeholders and their involvement in the configuration phases.

B. *Process and Tools*

To have the use case working on the AAL platform, configuration is needed, before processing a light switch request. The configuration framework splits into five phases for the integration of the use case in the AAL environment. Figure 2 gives an overview of these phases and their alignment to the stakeholders directly involved in the configuration (cf. Chapter II). They are overlapping in the phases if some interaction could be necessary during the use case integration. The configuration framework is implemented on top of the AAL platform (see Figure 2). The framework deploys the software, hardware and human resource components on the AAL platform though these phases:

The *development* phase contains all steps building software, hardware and human resource components.The two main stakeholders are the software and hardware developers. They implement the components and make them configurable. Therefore, the developers have to deal and extend with three types of ontologies (software, hardware, human profile; cf. Chapter III-A).

The use case “fall-prevention” can easily extend the ontology concepts to its needs (cf. requirement R1, R2, R3). In the configuration phase a tool is provided that allows for an easy packaging of the develop components to one use case. The tool also adds deployment information in a manifest.

The *design* phase is the place to find and combine software, hardware and human resources components to the needs and preferences of the elderly. The case manager uses the use case descriptions from the developers and the ontologies to combine different components to meet the requirements. This ensures a high flexibility and fault tolerance of the AAL platform.

The case manager can now easily design the AAL environment using the motion detection sensors (from the “fall-prevention” use case) also for the second “front door monitoring” use case (cf. R1).

The *installation* phase integrates the different components of the use case in the AAL platform. During the installation three different stakeholders are involved.

The case manager mainly coordinates the installation. For this task he is supported by the administrator of the AAL environment. During the installation, the use case manifest links to several information items (e.g. licences) and guides through the installation process. Based on the manifest information,

 ISBN: 978-1-61208-421-3

the compatibility of the use case to the rest of the system is checked. Also, the manifest describes the requierd information to install the ontologies delivered with the use case in the platform (e.g. path of the ontology profile, ID of the runtime instance loading the ontology). After the new ontologies are matched to the main system ontology and all dependencies are solved, the new ontology are resolved successfully.

The technician is responsible for the installation of hardware components. Mostly he will integrate new sensors and actuators in the environment and ensures that they are known in the AAL platform based in their semantic description. If the ontology concepts coming from the use case components matching the requirements, they can be added in the platform automatically. This phase addresses requirement R1 and R2.

The *customisation* adjusts the software, hardware and human resources components to the elderlys preferences. This is mostly done by creating instances of the previously installed ontologies. The installed hardware needs a representation on the software level to be used in the system. Also, other software components and human resources are set-up. Some of the information in the AAL environment can be reused in this step or can be used to automate the configuration by its machine readable description. Based on this it is possible to create fall back configurations to be fault tolerant.

For the example use cases the motion detection sensors and the light switch actuators have to be instantiated in the ontology. The case manager, possibly with help of the technician will ensure to set them up according to the ontology concepts. They will set the location, a label and other needed parameters. Furthermore, they may have to fill in additional information about the assisted person. After the hardware and human ontology is instantiated it could be necessary to link the software part to right hardware concepts or instances. All these information will adjust the use case to the assisted person (cf. R2, R3). Especially for the second use case reusing the motion sensors this is necessary (cf. R1).

A tool to creating or editing the ontology instances and to initialise the software part is located in the configuration framework.

The *maintenance* enables the deployer to easily control the environment. In this phase of the configuration process it is possible to edit software, hardware and human resources components by changing ontology instances. Of course, it is also possible to delete or uninstall components or even whole use cases. The maintenance phase can detect errors in the AAL platform and react accordingly to the fall back configurations or/ and inform the maintenance provider.

In the "fall-prevention" use case a ceiling light could be broken. With the common understanding of light sources and being organised in hierarchical concepts the platform and the maintenance provider can reconfigure the use case to use another light source located in the kitchen (cf. R4).

IV. Implementation

The configuration framework was implemented in Java, OSGi (Open Services Gateway initiative: http://www.osgi.org/) and a GWT (Google Web Toolkit: http://www.gwtproject.org/) derivate called Vaadin. It is part of the universAAL middleware running in the AAL environment accessible via browser from the outside. The first phase of the configuration framework is not part of the implementation because it provides guidance about developing configurable use cases and ontologies. The developer will just use his preferred IDE. The universAAL project provides Eclipse plug-ins to transform ontologies to Java and vice versa, to generate configuration files and to package the use case with its components and manifest. The implementation of the framework supports the installation, customization and maintenance phases. The implementation of the configuration framework consists of three main managers: The *Deploy Manager* installs and uninstalls use cases on the AAL platform is the purpose of the Deploy Manager. The *Configuration Manager* configures the AAL-service based on the configuration file delivered with the use case. The *Ontology Manager* enables the case manager to create or edit ontology instances (cf. Chapter III-A).

V. Discussion

Based on a detailed stakeholder analysis, a configuration framework was developed in order to flexibly manage AAL environments. It is integrated on the semantic middleware of the universAAL platform. This framework contains (i) a process guiding the stakeholders through the use case configuration and use case ontology development and (ii) tool support for the different phases in the process. With the configuration framework, the flexible management of AAL environments is supported.

This framework was installed and used in five living labs in Europe and in nine real homes. According to the iterative approach of the framework development, the first framework prototype "Jambi" was installed and tested in five living labs. The following example of the use case "nutritional advisor" with its ontology extensions shows how the evaluation was performed: Each living lab was asked to deploy this use case with the configuration framework:

1) Search and purchase the "nutritional advisor" use case in the online store
2) Download and install the use case
3) Customise the use case according to the needs of the assisted person
4) Verify correct integration of the use case in the AAL environment

All five living labs succeeded in deploying the use case with an average time of 36 minutes (4,80 standard deviation). One of the case managers deploying the use case needed support from the developer of the "nutritional advisor". The big time difference is caused by the varying prior knowledge of the case managers. Some of them had already deployed AAL use cases. They all agreed that the deployment process was too complex: "install an app involves a lot of steps (too many clicks!!!)" but they also agreed that it was helpful to have the guidance by the configuration framework. The goal in the next iteration was to try to automate configuration steps. Based on the problem that developers do not know which ontologies are available in the AAL environment, they started to deliver more and more own ontology extensions with their use case components. With the ontology editors help the deployer can easily reuse ontologies and their instances during the configuration. Of course, developers outside of the AAL environment were not aware of ontologies already deployed in the environment. After the first evaluation the refined framework was implemented

 ISBN: 978-1-61208-421-3

in the second prototype “Vaadin”. It was installed with the universAAL middleware in nine real homes running six use cases with 48 ontology extensions. The field trial can be seen as a proof-of-concept for the configuration framework. At the moment it is still on-going.

In the near future, the framework will be used in in the European research project ReAAL [19] which has the goal to deploy the universAAL platform with the configuration framework in 7000 real homes in Europe. The implementation of the framework the ontologies are available open source [20].

REFERENCES

[1] D. Cook and S. Das, Smart Environments: Technology, Protocols and Applications. Wiley, 2004.

[2] E. T. Ryan, D. R. Jacques, and J. M. Colombi, “An ontological framework for clarifying flexibility-related terminology via literature survey,” Systems Engineering, vol. 16, no. 1, Mar. 2013, pp. 99–110. [Online]. Available: http://doi.wiley.com/10.1002/sys.21222

[3] P. Wolf, “openAAL - the open source middleware for ambient-assisted living (AAL),” in Proceedings of the AALIANCE conference, 2010.

[4] M. Mikalsen, “Interoperability Services in the MPOWER Ambient Assisted Living Platform.” in Vol. 150 IOS Press, 2009, pp. 366–370.

[5] A. Schmidt, “SOPRANO Ambient Middleware: Eine offene, flexible und marktorientierte semantische Diensteplattform fr Ambient Assisted Living.” in Proceedings of the 2. Deutscher Kongress Ambient Assisted Living, 2009.

[6] A. Fides-Valero, “The PERSONA Framework for Supporting Context-Awareness in Open Distributed Systems.” in Proceedings of European Conference on Ambient Intelligence, 2008.

[7] C. Ardito, B. Barricelli, P. Buono, M. Costabile, R. Lanzilotti, A. Piccinno, and S. Valtolina, “An Ontology-Based Approach to Product Customization,” in End-User Development. Springer, 2011, vol. 6654, pp. 92–106. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21530-8_9

[8] M. Dong, D. Yang, and L. Su, “Ontology-based service product configuration system modeling and development,” Expert Systems with Applications, vol. 38, 2011, pp. 11 770–11 786.

[9] C. Cetina, P. Trinidad, V. Pelechano, and A. Ruiz-Cortes, “Customisation along Lifecycle of Autonomic Homes,” in 3rd International Workshop on Dynamic Software Product Line (DSPL09), 2009.

[10] F. Kawsar, T. Nakajima, and K. Fujinami, “Deploy spontaneously: supporting end-users in building and enhancing a smart home,” Proceedings of the 10th international conference on Ubiquitous computing (2008), Sep. 2008, pp. 282–291. [Online]. Available: http://dl.acm.org/citation.cfm?id=1409635.1409673

[11] M. Stumptner, “An overview of knowledge-based configuration,” AI Communications, vol. 10, no. 2, Apr. 1997, pp. 111–125. [Online]. Available: http://dl.acm.org/citation.cfm?id=1216064.1216069

[12] T. R. Gruber, “Toward principles for the design of ontologies used for knowledge sharing?” International Journal of Human-Computer Studies, vol. 43, no. 5-6, Nov. 1995, pp. 907–928. [Online]. Available: http://dl.acm.org/citation.cfm?id=219666.219701

[13] T. Zentek, A. Marinc, and A. Rashid, “Methods and Tools for Ontology-based Configuration Processes of AAL Environments,” in Ambient Assisted Living, R. Wichert and H. Klausing, Eds. Berlin Heidelberg: Springer, 2013, pp. 213–224.

[14] G. van den Broek, F. Cavallo, and C. Wehrmann, “AALIANCE Ambient Assisted Living Roadmap,” Tech. Rep., 2010. [Online]. Available: http://books.google.com/books?id=dDXPKWN61msC&pgis=1

[15] T. Zentek, C. Yumusak, C. Reichelt, and A. Rashid, “Which AAL Middleware matches my requirements? An Analysis of Current Middleware Systems and a Framework for Decision-Support.” in Ambient Assisted Living AAL, W. Reiner, Ed. Berlin Heidelberg: Springer, 2014.

[16] W. P. Tazari M.-R,Furfari F,Fides Valero Á,Hanke S,Höftberger O,Kehagias D,Mosmondor M,Wichert R, “The universal Reference Model for AAL,” in Handbook of Ambient Assisted Living, J. et al. Augusto, Ed. IOS Press, 2012, pp. 610 – 625.

[17] B. Bergvall-kåreborn, B. Bergvall, M. Holst, and A. Stå hlbröst, “Concept Design with a Living Lab Approach,” Proceedings of the 42nd Annual Hawaii International Conference on System Sciences, HICSS (2009), 2009, pp. 1–10.

[18] V.-p. Niitamo, S. Kulkki, M. Eriksson, and K. A. Hribernik, “State-of-the-Art and Good Practice in the Field of Living Labs,” 2006.

[19] Project ReAAL, “ReAAL,” http://www.cip-reaal.eu/home/, 2015, [Online; accessed 19-Juni-2015].

[20] Project uAAL, “universAAL,” http://forge.universaal.org/svn/uaaltools/trunk/uCC/ ontology.universaal.org, 2015, [Online; accessed 19-Juni-2015].

 ISBN: 978-1-61208-421-3