openAAL[1] - the open source middleware for ambient-assisted living (AAL)

Peter Wolf[1,]*, Andreas Schmidt[1,]*, Javier Parada Otte[1], Michael Klein[2], Sebastian Rollwage[2], Birgitta König-Ries[3], Torsten Dettborn[3] and Aygul Gabdulkhakova[3]

1 FZI Forschungszentrum Infromatik, Karlsuhe, Germany
2 CAS Software AG, Karlsruhe, Germany
3 Friedrich-Schiller-Universität, Jena, Germany

E-Mails: peter.wolf@fzi.de ; andreas.schmidt@fzi.de; parada@fzi.de; michael.klein@cas.de; sebastian.rollwage@cas.de; birgitta.koenig-ries@uni-jena.de; torsten.dettborn@uni-jena.de; aygul.gabdulkhakova@uni-jena.de

* Authors to whom correspondence should be addressed; Peter Wolf: Tel.: +49 721 9654-718 / Andreas Schmidt: Tel.: +49-721-9654-732

Abstract: openAAL is a joint open source initiative by FZI Research Center for Information Technologies, Friedrich-Schiller-University of Jena and CAS Software AG. It represents a flexible and powerful middleware for ambient-assisted living (AAL) scenarios and is based on research results of several German and international projects including the SOPRANO[2] Integrated Project. The openAAL platform enables easy implementation, configuration and situation-dependent provision of flexible, context-aware and personalized IT services.

Keywords: openAAL; SOPRANO; middleware, AAL

1. Introduction

AAL solutions cannot be off-the-shelf products. Too diverse are the needs and living conditions of elderly people, and moreover, they change over time, sometimes very quickly. On the other hand, we can make AAL solutions only affordable if they are not just engineered for a single case. As a consequence, technical solutions need to be flexible and adaptable to individual and changing needs. As other examples in complex domains show, this kind of flexibility cannot be achieved by a single vendor, but rather by a living market ecology in which different actors can bring in their strengths. At the core of such market ecology you need an appropriate common technical platform as an enabler that allows for the integration of their services, products, and experiences.

[1] www.openaal.org
[2] www.soprano-ip.org

Towards that end, we have developed the openAAL middleware (http://openaal.org) as part of the SOPRANO IP and provided an open source solution to the emerging community and market. The openAAL middleware defines such a framework on top of OSGi that allows for easy integration of and communication between services. Additionally, openAAL provides generic platform services like context management for collecting and abstracting data about the environment, workflow based specifications of system behaviour and semantically-enabled service discovery. Framework and platform services are loosely coupled by operating and communicating on shared vocabulary (ontology).

In the following sections the role of OSGi as base technology, the three main components that define the system's architecture and platform services (viz. Context Manager, Procedural Manager and Composer, as seen in Figure 1.) as well as the shared openaal ontology are shortly presented (a more detailed overview can be found in (2) and (3)).

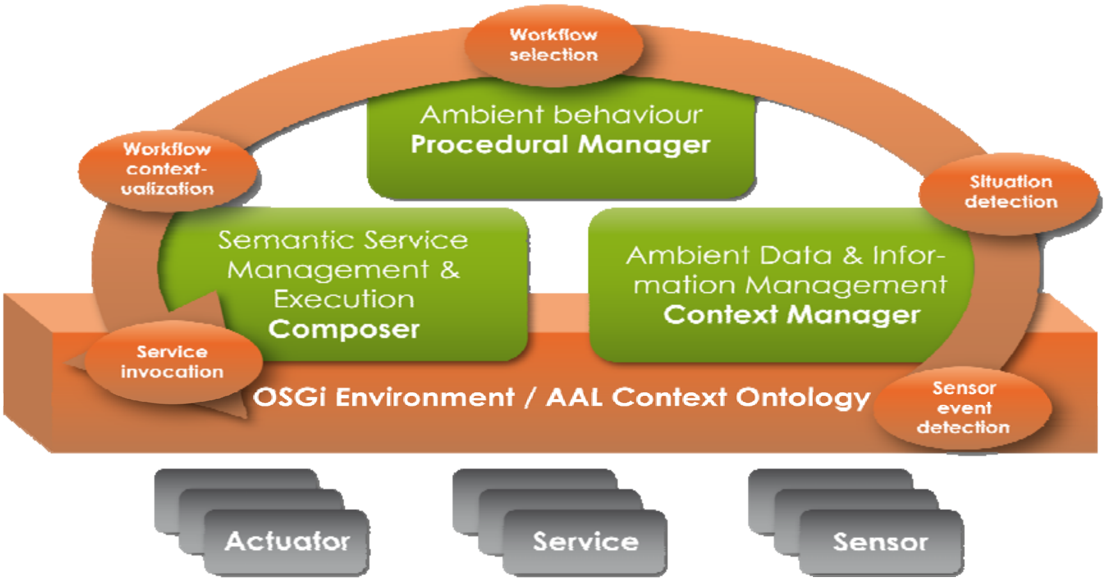
2. Architecture

The openAAL middleware is based on the widespread middleware framework OSGi. As a service-oriented framework **OSGi** reduces the coupling between components, for example by:

- specifying a new module entity called bundle that has a framework independent lifecycle,
- automatic management of module dependencies,
- natural support of dependency injection and
- enabling event-based communication.

In openAAL OSGi serves as a base layer for the three main components: Context Manager, Procedural Manager and Composer. Hence, all interaction between these components and services is managed by OSGi.

Figure 1. openAAL architecture



Context Manager provides an ontology-based information storage that captures sensor information and user input (see (1) for detailed overview). Internally, a blackboard architecture facilitates the use of different algorithms to derive situations of interest (e.g., based on rules, logical reasoning, Bayesian networks, etc.). Complex sensor settings and situation detection algorithms may generate conflicting data (e.g., on the position of a person). With the help of a conflict resolution component, the Context Manager can provide a conflict-free view for services, while complex situation detection algorithms may use all the collected data including inconsistencies.

Especially, the openAAL framework supports context reasoning from a low-level sensor-based model to a high-level service-oriented model. The low-level and the high-level model form the openaal ontology. Modelled in a state-based manner the low-level ontology provides and defines statements that can be used to describe sensor and actuator states. The high-level ontology focuses on the immediate environment of the AP and provides a vocabulary to define preferences, location, activities and situations of the AP. These two different models enable easy integration of sensors and AAL services.

Based on RDFS[3]-like syntax, statements composed of subject, predicate, object as well as domain specific meta-information can be used to describe sensor-states and situations. The meta-information contains timestamps as well as confidence information to enable more complex reasoning about history and uncertain data.

The **Procedural Manager** manages and executes easy to define and installation independent workflows which react to situations of interest. Those workflows are specified by means of BPEL-processes extended with constructs that allow for synchronous and asynchronous communication with the Context Manager. In general, these workflows define the system's reaction to certain situations as identified within the Context Manager and can be defined to automatically resolve critical situations or inform a responsible person. Hardware independence is achieved by defining the system reaction upon specific installation independent situations of interest instead of concrete sensor events. Furthermore, service calls are specified as abstract service requests which are independent of specific service offers and, therefore, software and hardware that is currently available. This kind of installation independence enables reuse of configuration data over several different installations. Also, otherwise complex workflows tend to become simpler, since service concretisation and situation detection are shifted into other components that can be configured separately. Workflow specification should be done collaboratively by people with extensive general knowledge about the AAL domain and people that share the life of the assisted person(s).

The **Composer** analyses services which are available in a certain installation and selects and combines those services to achieve the (abstract) service goals as outlined within the Procedural Manager. Internally, service matchmaking mechanisms are based on the DIANE Service Description (DSD) framework (see (4)). openAAL adds the notion of virtual services which can be used to bridge

[3] Resource Description Framework Schema: http://www.w3.org/TR/rdf-schema/

the semantic gab between abstract service requests and concrete service goals. Virtual services offer abstract services and upon concretisation orchestrate concrete services to achieve a maximal adaptation to the user's needs in the current situation.

3. Projects

openAAL is currently part of several EU- and German nationally funded projects. The next section gives a short overview about the currently 5 projects that use and contribute to openAAL in terms of topics and goals as well as their (intended) use of the openAAL.

SpeedUp[4] project aims to develop a framework that provides IT-support for the cooperation and communication of emergency management organizations. Up to now, such cooperation is achieved more or less manually by orally exchanging information or exchanging print outs. In the case of scenarios such as mass casualty incidents, however, personal consultations or the procurement and dispatch of documents are virtually impossible to manage. Processes are inefficient and precious time is lost as a result: factors that, in the worst case, raise the number of victims. Research group from University of Jena will be investigating the possibility to dynamically map emergency management processes to varying underlying resources. Resources and requirements will be semantically described in terms of semantic web services and then automatically discover, compose, bind, and configure services to achieve a given goal by using the Procedural Manager and Composer functionalities.

The aim of the "**MOPS**[5]" project is the development of solutions for the transfer of formerly static business processes into the world of mobile data devices. The project under the title "Adaptive Planung und sichere Ausführung mobiler Prozesse in dynamischen Szenarien (MOPS)" (Adaptive Planning and Secure Execution of Mobile Processes in Dynamic Scenarios) is funded by the "Thüringer Aufbaubank" and began on the 06.08.2009. As one goal, the project will investigate the application of openAAL in the areas of mobile data devices.

FZI living-lab AAL[6] is a project funded by the state of Baden-Württemberg (Germany). The still ongoing project provides lab-environment resembling a technology-enhanced flat where an older person could live in. The lab serves as a showcase and discussion forum for companies and researchers on topics connected to AAL. Furthermore, it can be used as integration platform for AAL-related technologies and services. openAAL is used as underlying infrastructure and several use cases demonstrating reminder and safety functionality are already implemented.

SOPRANO is an EU-funded project that aims at supporting independent living of older people in their own home. Main topics within the projects are design and use of a new evaluation methodology for AAL-technologies tailored to AAL-domain specific requirements, investigation of market ecology and user needs and development of a basic infrastructure to support user, developer and service provider needs. The current version of openAAL was mainly developed as part of SOPRANO's technical activities.

[4] http://www.speedup.uni-jena.de/

[5] http://mops.uni-jena.de

[6] aal.fzi.de

universAAL, an EU-funded project that has recently started, aims at collecting scenarios, requirements and technical solutions from various projects in order to consolidate and standardize the AAL middleware platform. openAAL together with SOPRANO is one of several “input projects“ that will be taken into account during consolidation.

4. Conclusions

With openAAL, we have created an ontology-based middleware that allows for the required flexibility and adaptability in AAL scenarios. It lays the foundation for an AAL service market ecology in which different market participants can contribute with their experience:

- hardware sensor manufacturers can plug in their sensors into the system by providing a simple semantic description of their data,
- added-value service developers can develop new methods to analyze sensor-level data and detect situations, e.g., for activity detection, and plug in into the context manager
- tele- and healthcare consultants can develop system workflows for an adequate system reaction to certain situations, based on the domain knowledge
- care providers or informal carers can configure the system behavior by activating and deactivating procedures and by providing information about preferences and impairments of the assisted person(s)

As one of the first AAL platform infrastructures, openAAL has been released as open source. And it is currently being used and developed further in several initiatives.

References and Notes

1. Peter Wolf and Andreas Schmidt and Michael Klein, **Applying Semantic Technologies for Context-Aware AAL Services: What we can learn from SOPRANO**; Workshop on Applications of Semantic Technologies 09, Informatik 2009
2. Peter Wolf and Andreas Schmidt and Michael Klein, **SOPRANO - An extensible, open AAL platform for elderly people based on semantic contracts**; 3rd Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI’08), 18th European Conference on Artificial Intelligence (ECAI 08), Patras, Greece
3. Michael Klein and Andreas Schmidt and Rolf Lauer, **Ontology-Centred Design of an Ambient Middleware for Assisted Living: The Case of SOPRANO**; Towards Ambient Intelligence: Methods for Cooperating Ensembles in Ubiquitous Environments (AIM-CU), 30th Annual German Conference on Artificial Intelligence (KI 2007), Osnabrück, September 10, 2007
4. Michael Klein, Birgitta König-Ries, Michael Müssig, **What is needed for Semantic Service Descriptions - A Proposal for Suitable Language Constructs;** International Journal of Web and Grid Services 2005 - Vol. 1, No. 3/4