Building AAL Services Using the MILEO Context Server

Manfred Wojciechowski [1,*], Wolfgang Deiters [1]

Fraunhofer Institute for Software and Systems Engineering / Emil-Figge-Str. 91, 44227 Dortmund, Germany

E-Mails: manfred.wojciechowski@isst.fraunhofer.de, wolfgang.deiters@isst.fraunhofer.de

Tel.: +49-231-97677-0; Fax: +49-231-97677-199

Abstract: Ambient Assisted Living (AAL) services provide intelligent and context aware assistance for elderly people in their home environment. This domain puts special requirements on context modeling that are not in the scope of current context modeling approaches. These include the separation between sensor provided context information and service specific context model, the provision of service-type specific context models and an end user view on context models. The MILEO context server meets these requirements through the provision of a three layered context model, where each layer is focused on one of these aspects. MILEO provides an XML-based description language, which can be used to implement the context model. It also provides a graphical user interface component that supports end user context modeling. We describe how this can be applied to build an AAL service.

Keywords: AAL service, context server, three layered context model

---

1. Introduction

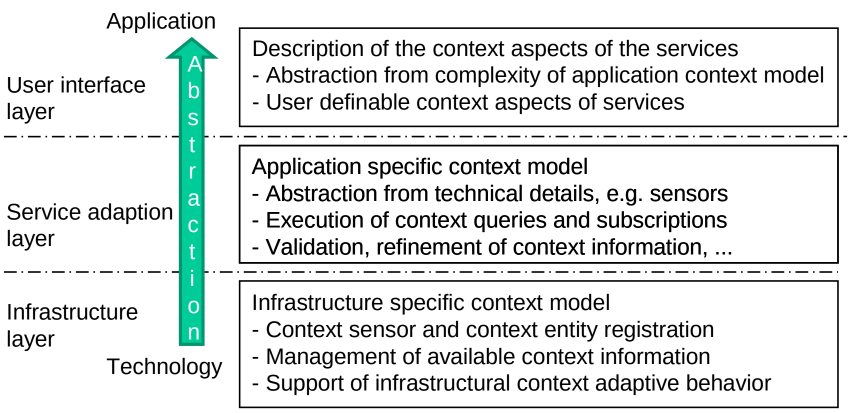
‚Ambient Assisted Living' (AAL) aims at extending the time that the elderly can live in their home environment independently. A smart home environment integrates into the living space of the inhabitant and provides services that help to increase their autonomy and gives assistance in different activities of daily life. Key technologies for AAL services can be found in the research areas of 'home automation' and 'ambient intelligence.' AAL services provide intelligent and context aware assistance for elderly people in their home environment. We follow Dey's [1] definition in which context is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." Examples of context aware AAL services can be found in [2]. In the project 'SmarterWohnen' [3], we have implemented and tested a number of AAL services together with a local housing company. These services include intrusion detection, water and gas leakage detection, health related services and other various home automation services.

From our experience this domain puts special requirements on context modeling that are not in the scope of current context modeling approaches. We have developed the MILEO context server, which supports these requirements. In the following we will give a short introduction to our context server. We will start with the special requirements and the resulting three layered context model. Then we give an overview of the architecture identifying the main components of the MILEO context server. Finally we introduce the XML-based description language and demonstrate how to build an AAL service.

2. Three Layered Context Model

AAL services have to rely on the context infrastructure that is available in a smart home environment. The service developer is not the owner of that environment and therefore cannot define and build on its own infrastructure. This infrastructure has to be shared by many different AAL services. Therefore there is the need to separate between the individual smart home context infrastructure and the context models used by the different AAL services. A second requirement is the provision of service-type specific context models. As described in [4] AAL services can be categorized according to their application domain. We distinguish between the following AAL service types: health, security, comfort, social and economic services. Each of these service types has a different scope on what context information is relevant, e.g. health related services focus on vital parameters of the inhabitant and nursing staff. Therefore different service-type specific context models have to be provided by the same context infrastructure. Finally the end user has to be considered in designing AAL services. Context aware services within a smart home environment have to behave according to the inhabitant’s expectations. Developers of such services cannot foresee the behavior that is expected by the end user. Therefore it is important to allow the inhabitant to define the context aware behavior of his services. Some concrete requirements in the end user aspect have been identified in [5].

Figure 1. Three layered context model

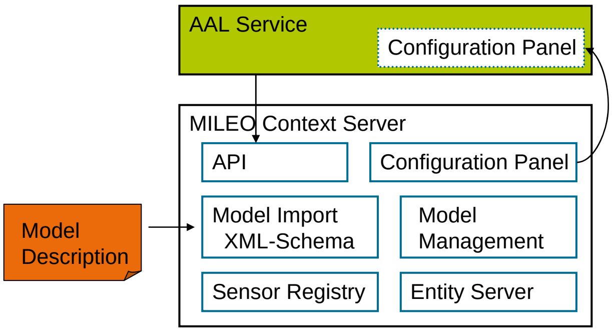


As a result of these requirements we have organized our context modeling approach into three separate layers with a specialized point of view on the infrastructure, the service adaption and user interaction. An overview of this three layered context model is given in figure 1. The context model on the infrastructure layer represents the existing infrastructure resources, e.g. context sensors and the service independent view on the smart home environment. It manages the different entities and the context information that is available. It is used for the integration of new context sensors. The context model on the service adaption layer builds on top of the infrastructure layer. It provides the different service-type specific context models. Using given mapping functionalities these models can be defined for each service type and enriched with additional semantics, e.g. the role of a person. Another model element on that layer is the definition of context spaces that represent situations observable through context information, e.g. emergency. This layer is used to provide the context-adaptive features of the AAL services. The context model on the user interaction layer builds on top of the service adaption layer. It is focused on the visualization of context aspects to the inhabitant and allows the definition of the context-adaptive behavior of the AAL services by the end user. It abstracts from the complexity of the underlying technical context models. The core concept of this layer is the usage of predefined situation taxonomies that relate to the context spaces of the service-type context models. The situation taxonomy can be used by the inhabitant to identify and select the desired context aware behavior by navigation. After a taxonomy element has been selected the inhabitant can apply refinement functions to further specify the desired behavior, e.g. the selection of a specialized context entity. A detailed description of this layer is given in [6]. A description of its application is given in [7].

2. Architecture of the MILEO Context Server

The following figure gives an overview of the components of the MILEO context server.

Figure 2. Architecture overview



Concrete context models can be defined using an XML-based model description language. This language is based on a meta model, which defines the relevant concepts on each of the layers. Elements of the meta model are for example ‘Entity’, ‘Relation’, ‘Attribute’ and the ‘Context Space’. A model description, e.g. a ‘health’ context model, is read by the ‘Model Import’ component and automatically transformed into its implementation, which is based on a relational database management system enhanced with semantic features. The ‘Entity Server’ is responsible for the registration of concrete context entities and the provision of identifiers, e.g. inhabitant ‘John Doe’ with the ID ‘person-01’. The types of the entities that can be registered are defined in the infrastructure layer of the context model. The ‘Sensor Registry’ enables the ad hoc integration of context sensors using a sensor description language. This builds on the infrastructure layer for the description of context information that can be provided by the sensors. An API provides access on the service-type specific context models of the service adaption layer. An AAL service can use these methods to query for existing context information, but also to subscribe for context events. The 'Configuration Panel' is a component that can directly be integrated into the user interface of an AAL service. It has access to the user interface layer of the context model and provides the interaction forms that can be used by the inhabitant to define the expected context-aware behavior of the service.

3. Building of an AAL Service

In the following we describe the usage of the MILEO context server for the building of an AAL service by a simplified example of a vital parameter supervision service, which will send notifications when a critical value is exceeded. There are two phases for the realization of such services, which we describe in the following sections.

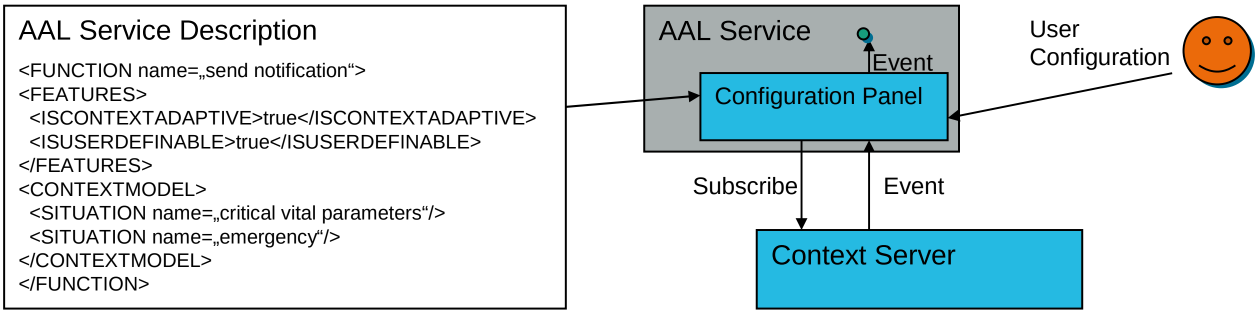
3.1. Instantiation of the MILEO Context Server

The objective of the first phase is to provide the MILEO context server as a basis for the context-aware behavior of an AAL service. In a first step the context models on each of the three layers are defined. On the infrastructure layer we the context entities according to their common properties. For example we can define an abstract entity 'movable object', which describe entities with the current location as an associated context attribute. A specialized entity 'person' additionally contains attributes regarding different vital parameters and its role, e.g. 'inhabitant' or 'nurse'. On the service adaption layer we define a context model for the service type 'health'. Therein we define the context entity 'patient', which is mapped to the entity 'person' of the infrastructure layer with the attribute 'role' is 'inhabitant'. The attributes of this entity are limited to the vital parameters. Then the context spaces are defined, e.g. 'high blood pressure', described as the vital parameter 'blood pressure' of the context entity 'patient' is higher than '140 mmHg'. On the interaction layer then the situation taxonomy is defined. It includes a situation 'critical vital parameter' with 'high blood pressure' as one specialization. These models are described using the XML-based model description language and imported into the MILEO context server. In a next step the entity server is used to register entity 'John Doe' of type 'person'. Finally the blood pressure sensor can be registered in the sensor registry and assigned to the attribute 'blood pressure' of the entity instance 'John Doe'.

3.2. Implementation of the AAL Service

We distinguish between hard wired and user definable context-aware behavior of AAL services. Hard wired behavior can be implemented using the event-condition-action paradigm within the implementation of the service, where the event part is supported by using the event subscription method of the context server API. For user definable context-aware behavior the event and the condition part are realized outside of the service by using the 'Configuration Panel'. This component can be integrated into the user interface of the AAL service and handles all user interaction concerning the definition of the desired context-aware behavior. A service description has to be provided together with the component. This contains a list of context-aware functions, e.g. 'send notification', and parts of the situation taxonomy, that can be used for the configuration of the function. This description is used by the 'Configuration Panel' to provide the configuration dialogue based on the interaction layer of the context model, e.g. navigate through the taxonomy and make adjustments on the values of the critical vital parameters.

Figure 3. Implementation of the end user definable context-aware behavior



4. Conclusions

Using the MILEO context server we can support different types of AAL services using the same context infrastructure. The development of an AAL service consists of two steps - the instantiation of the context server and the implementation of the service itself. In case of end user definable context-aware behavior the implementation is limited to the core functionality. All context-dependent aspects are realized descriptively, using the XML-based model description language and the service description.

References and Notes

1. Dey, A. Providing Architectural Support for Building Context-Aware Applications. PhD Thesis, Georgia Institute of Technology, 2000.
2. Meyer, S.; Rakotonirainy, A. A Survey of Research on Context-Aware Homes. Proc. of the Australasian information Security Workshop on ACSW Frontiers 2003, vol. 21, pp. 159-168.
3. Meis, J.; Draeger, J. Modeling automated service orchestration for IT-based homeservices. Proc. of the Int.Conf. on Service Operations and Logistics and Informatics SOLI '07, pp. 155-160.
4. Nehmer, J.; Becker, M.; Karshmer, A.; Lamm, R. Living assistance systems: an ambient intelligence approach. Proc. of the 28th Int. Conf. on Software Engineering 2006, pp 43-50.
5. Kleinberger, T.; Becker, M.; Ras, E.; Holzinger, A; Müller, P. Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces”. Universal Access in HCI, LNCS 4555, 2007, pp. 103–112.
6. Wojciechowski, M.; Xiong, J. A user interface level context model for ambient assisted living. Proc. of the Int. Conf. on Smart Homes and Health Telematics 2008, LNCS 5120, pp. 105-112.
7. Wojciechowski, M. End User Context Modeling in Ambient Assisted Living. Int. Journal of Advanced Pervasive and Ubiquitous Computing, Vol.1 (2009), No.3, pp.61-80.