Towards Mobility Support in Smart Environments

Daniel Retkowitz Ibrahim Armac
Manfred Nagl
Department of Computer Science 3 (Software Engineering)
RWTH Aachen University, Ahornstr. 55, 52074 Aachen, Germany
{retkowitz|armac|nagl}@i3.informatik.rwth-aachen.de

Abstract

Smart environments are subject to intensive academic and industrial research. Many of these research projects deal with challenges such as heterogeneity, personalization and context-awareness. However, most of them assume smart environments to be insular places. Considering users visiting different environments in daily life, this assumption becomes unrealistic.

Mobile users wish to use personal functionality in their home environment as well as in other environments they visit in daily life. In this paper we describe different realization patterns to implement services for smart environments. The aim is to support personalization of services and mobility of the users. Depending on the application, different realization patterns are preferable. Furthermore, we describe how our prototype implementation supports the different patterns.

1. Introduction

In this paper we describe our approach on supporting personalization and mobility in smart environments, in particular smart homes, which we call *eHomes*. These are environments equipped with devices which are usually connected to a hardware platform called *residential gateway*. This gateway runs software services to realize value-added functionality across multiple devices.

A specific challenge in realizing eHomes is to deal with mobility. One kind of mobility is given when users move from one location to another one (*in-home mobility*). In most cases a *location* is a room in an eHome. However also larger areas of an eHome that comprise several rooms or a part of a larger room can be modeled as one location. Another kind of mobility is given when users move from one eHome to another eHome (*inter-home mobility*). In this case, the term eHome is used in a broader sense meaning also environments such as a hotel, work place etc. Furthermore, also the mobility of devices and changing user preferences have to be taken into account to support dynamics in eHomes.

Considering inter-home mobility, the important question arises how to support users in personalizing visited environments. For this purpose we pursue a *client side personalization* approach. This approach is based on the assumption that every user carries a smart mobile device which can support the user in personalization tasks. Nevertheless, not all services need to be personalized. Therefore we distinguish *personal services* which adapt their functionality to user preferences and *non-personal services* which provide functionality at a specific location in an eHome or for an eHome as a whole.

Mobility requires a dynamic eHome system that reacts on changes and adapts to the new situation. In our project we developed a configuration approach that especially supports the requirements becoming apparent in dynamic scenarios considering mobility of users and devices. We analyzed different patterns for realizing services in mobile scenarios, where to apply them, and what implications they bring along.

We will describe our approach on supporting personalization and mobility in Section 3. Before that, we introduce our eHome system model in Section 2. In Section 4, we will discuss related work. Finally, we will conclude the paper with a summary and an outlook to future work in Section 5.

2. System Model

In future smart environments we assume different usage scenarios. Typical services provide functionality from the domains of comfort, entertainment, communication, security, health care, or time and energy saving. An *eHome service* implements a certain functionality, which is provided either directly to the users of

[RAN09] D. Retkowitz, I. Armac, M. Nagl
Towards Mobility Support in Smart Environments
In: Proceedings of the 21st International Conference on Software Engineering and Knowledge Engineering (SEKE 2009),
Boston, Massachusetts, USA, pp. 603-608. Knowledge Systems Institute Graduate School, Illinois, USA, 2009
www.se-rwth.de/publications

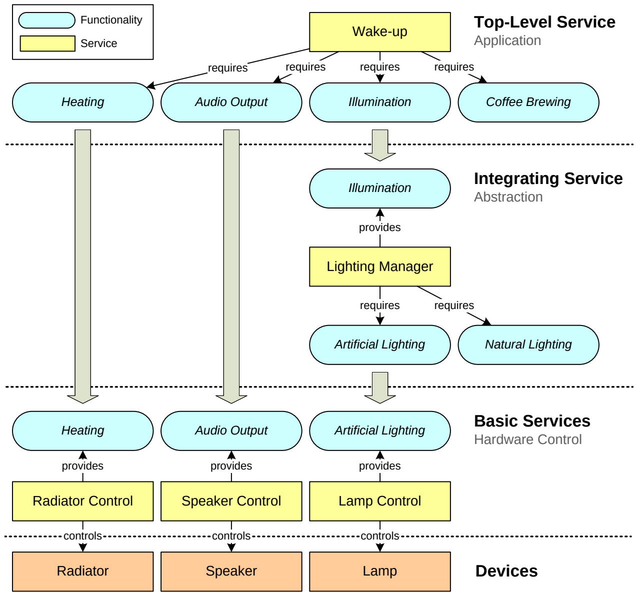

Figure 1. Layered Services

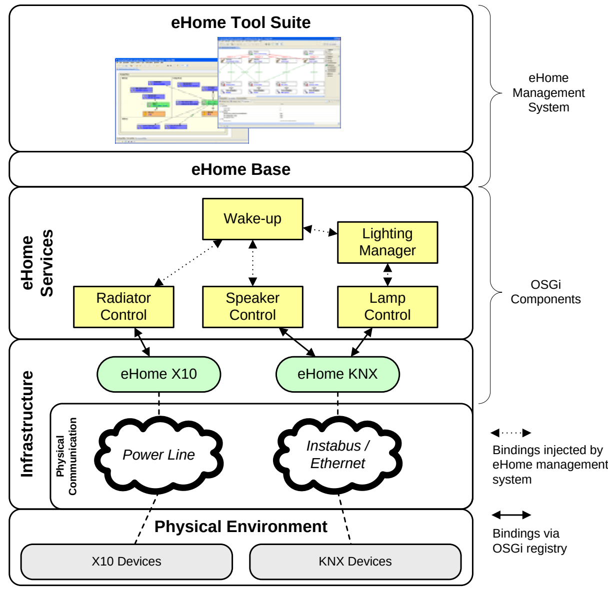

Figure 2. System Overview

an eHome or to other services. We distinguish between three different service types: top-level, integrating, and basic services. Services may rely on functionalities provided by other services on a lower level of abstraction. This leads to a layered service architecture.

2.1. Service Layers

Figure 1 shows a wake-up service as an example. This type of service is called *top-level service* since it provides its functionality directly to the user. The wake-up service requires other functionalities to operate, which have to be provided by services on a lower abstraction level. In this case heating is required to increase the room temperature before wake-up time. Audio output is required to play some wake-up sound or music. Coffee brewing functionality is used to prepare coffee after wake-up, so the person does not have to wait during the brewing process. Illumination functionality is used to slowly increase the illumination level at the location of the person to wake-up. This allows for a comfortable wake-up procedure. Illumination is here controlled by an intermediate lighting manager service, which provides illumination based on artificial or natural lighting. In case there is bright sunlight outside, the roller blinds can be used to control the illumination level. In other cases, especially during the night of course, artificial lighting is used for this purpose. On the lowest level of abstraction basic services are used to provide access to the available hardware in the eHome, e. g. to control radiators, speakers, or lamps.

A service composition like in the wake-up scenario is depending on the available hardware and the current status of the environment. Since changes occur frequently at runtime, the service composition has to be adaptable. In our approach the configuration of the eHome is managed by a service-oriented middleware running on the residential gateway. In the next section we describe the basic system architecture we apply. Details of the configuration mechanism beyond the scope of this paper are described in [7].

2.2. Service Gateway

The *service gateway* is a software platform for executing eHome services. It is running on the residential gateway, the central hardware unit of the eHome, which is in control of all hardware usable by eHome services. This means that the services are not distributed in terms of their execution. Only the device hardware that is controlled by driver services is distributed. Nevertheless, a service can be bound to a specific location. This means that the provided functionalities of that service take effect at this specific location.

The system configuration is also managed centralized by the service gateway. We pursue an approach based on a global view of the eHome and its current environment status. This way global context information can be taken into account for service composition, e. g. the location of persons and devices in specific rooms. Global knowledge of the environment is required for a meaningful service composition in many typical cases.

A simple example is the requirement to bind a resource from the specific location a service is associated to. This requirement can only be formulated if we have a concept of different locations in the first place and if we know which resources are available at this location at a given time.

Figure 2 shows an overview of the service gateway architecture we apply and how the service gateway is connected to the devices in the eHome. The top-most layer is a graphical interactive tool called *eHome Tool Suite* that is used to monitor and administrate the eHome system at setup and during runtime. A service specification editor is also integrated into this tool.

The data model and application logic for managing the eHome system is realized on the underlying eHome Base component. The data model is used as a representation of the current state of the eHome system comprising the physical structure of the eHome but also the dynamic state, i. e. the currently present users and their positions inside the building, the active devices, and the running services and their composition. The application logic of the eHome Base component implements the control capabilities to manage the eHome system configuration, i. e. the composition of services, service parameterization, and other runtime aspects. Furthermore, the deployment of system configurations is performed by this component.

The deployed instances of eHome services are executed on the next layer. This is controlled as described above by the eHome management system, i. e. the eHome Tool Suite and the eHome Base component. Services are composed according to their required and provided functionalities, the current environment status, and the user's specific requirements. Depending on the type of the used hardware a corresponding infrastructure is used for accessing this hardware, e. g. X10 devices or KNX (ISO/IEC 14543) devices. These devices are connected to the residential gateway via the eHome's power line or KNX network infrastructure, respectively.

3. Service Realization Patterns

In this section, we will introduce a classification of possible patterns for realizing eHome services, discuss our approach on mobility and configuration support, and describe implementation details.

In Section 2 we have described the different service layers. In this section we will discuss different realization patterns of top-level services regarding their bindings to locations or persons.

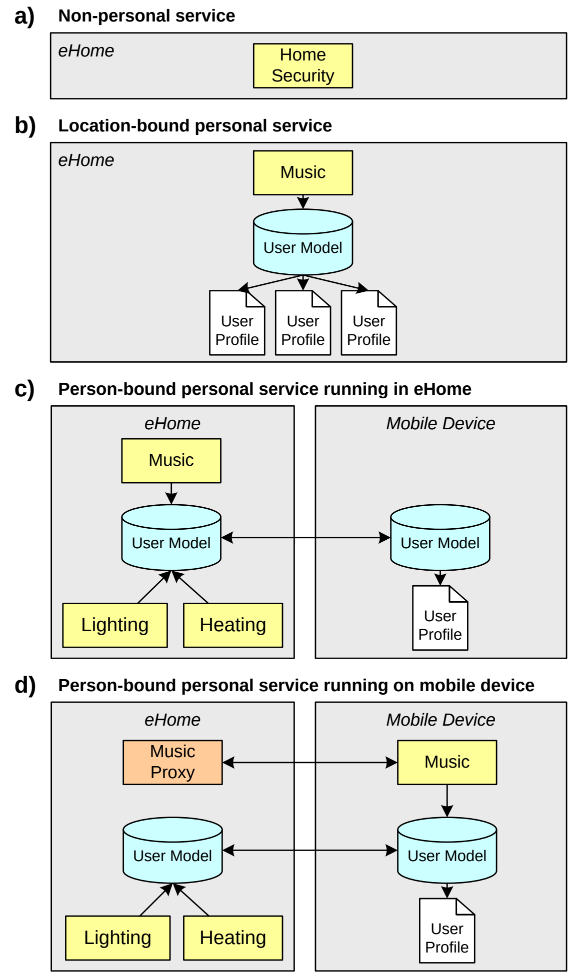

Figure 3. Different realization patterns for eHome services

3.1. Non-personal Services

Non-personal services are usually bound to locations and are not related to any specific person, as depicted in Figure 3a. Such services provide general location-based functionality, e. g. a home security service that detects intrusion or fire and raises an alarm. They obviously do not require any personal data to operate and are only related to spatial context. Any non-personal service is usually bound to a specific location where its functionality will take effect.

3.2. Personal Services

Personal services are related to individual persons and therefore require personal data, provided by user profiles. Besides the spatial context they also relate to personal context, e. g. a music service depends on the user's location and music preferences. We have developed a *user model* which holds personal data and provides it to personal services by a unified interface. There are different ways to realize personal services.

A *location-bound realization* is similar to the realization of non-personal services in that the service is associated to a specific location. In contrast to non-personal services now the users present at this location are taken into account. Depending on the presence of users the service accesses the user model which provides personal data for the different users, as shown in Figure 3b. Based on user preferences, the service personalizes its functionality for a specific user. However, if the user leaves the service's location and some other user arrives the service will personalize its functionality for this new user.

On the one hand, this realization allows to implement specific mechanisms for personalization and conflict resolution in the service implementation, e. g. in case of the music service it is possible to search for some common music preference that all currently present users have in common. Alternatively, it is also possible to use priorities for each user or to keep playing music for the user who arrived first at the service's location. On the other hand, this realization requires to decide where to run the service in advance and individual service instances are needed for all locations. Furthermore, it requires to implement person management and conflict resolution mechanisms for each service again and again. This leads to a lot of implementation redundancy and contradicts to reuse. In addition, while every service implements its individual mechanisms, this can lead to non-uniform behavior.

Another pattern is a *person-bound realization* which means that the service instance is no longer bound to a fixed location but to a specific user. Now, the service instance "follows" the user. This means that the service is bound to the user's current location at any time and this association is changed according to the user's movement, thereby supporting in-home mobility. Since the service is now related to one specific user it only needs to access personal data for this user. In this realization person management and conflict resolution have to be handled by the middleware.

Yet, we have described, how we support in-home mobility by the different kinds of service realization. However, a person-bound realization is also the basis supporting inter-home mobility. Details of inter-home mobility support will be discussed in the following section.

3.3. Mobility Support

Considering inter-home mobility, we want to enable hassle-free access to visited environments, while allowing users to keep their preferences for personal services across multiple environments. Therefore the visited environment must have access to these preferences. There are several ways of how to provide personal data to visited environments.

One way would be to store the users' profiles in a central (Internet-based) repository. Every visited eHome would then get access to the profiles of its "logged-in" users. A major disadvantage of that solution is that all the personal data, including sensitive data such as medical data, is managed by the central repository and requires the users to trust this repository. However, many users may not be willing to do this. Another way would be to interconnect the visited eHomes and the user's "home" environment where the personal data is stored. The downside of this approach is that it requires to tell the visited eHomes where one is coming from. This makes anonymity of users difficult and conflicts with the protection of privacy. The third way would be to store personal data on a mobile device. Taking along personal data on a mobile device allows a user to release his preferences to visited eHomes on demand. This is depicted in Figure 3c.

We went for the last alternative as it does not have the disadvantages mentioned above. We refer to this approach as "client side personalization" [1]. A similar approach is also suggested in [4].

The user model is responsible for exchange of personal data between a user's mobile device and the visited environment, see again Figure 3c. This includes transfer of data to the environment during log-in, synchronization during the session, and deletion after log-out. For more details about the user modeling component, including privacy aspects, see [3].

Up to know, we assumed that functionalities which a user desires are realized by already running services in the visited environment. In this case, it would be sufficient only to transfer the necessary personal data to the environment. However, there might be situations, where the visited environment does not run the wished services. Now the question arises how a user still can be served with the desired functionalities.

We have extended our approach so that a user can take along also personal services, in addition to personal data, and execute them on his mobile device when needed. An example is shown in Figure 3d. On the residential gateway a proxy service is deployed which encapsulates the connection to the mobile device where the actual service is running. Usually, the service needs to be bound to basic or integration services running in the eHome. Proxy services encapsulating the connection to these (remote) services, e. g. Speaker Control, will be generated on the mobile device and bound to the actual service on the mobile device, e. g. Music. This realization has the same effect from the user's point of view as if the mobile service would be running on the residential gateway.

Beside providing the user his desired functionality, this approach has another important advantage. Whenever a personal service is executed on a user's mobile device, the necessary personal data can be kept confidential on this mobile device. Thus, the amount of personal data transferred to the environment is reduced and the privacy protection enhanced. This is an important requirement, especially when moving through different and possibly unknown environments. However, this approach has also some disadvantages. It implies e. g. higher communication effort regarding service interaction between mobile device and the residential gateway. Furthermore, the energy consumption of the mobile device increases.

3.4. Configuration Support

As described in Section 2, top-level services are usually bound to further services, which can be integrating or basic services. There are several events which can affect a configuration.

One type of these events occurs when a device appears to a location. If the basic service controlling the new device is required by some other service bound to the same location, it will be bound to this service. In case of a disappearing device, it might happen that a location-bound service will be marked as invalid after its required service is unbound. These actions are similar for location-bound and person-bound services.

Other events occur when a person moves from a location to another one. In these cases, only personal services are affected. The case of location-bound personal services has been described previously in Section 3.2. In case of person-bound personal services, the situation gets more complex. Here, we have to distinguish two situations. If the service is running on the residential gateway, the personal service first will be paused. Next, the bindings of that service to basic services bound to the location which the user has left will be released. Then, the personal service will be bound to basic services providing the same functionality bound to the location which the user has entered. Finally, the service will be restarted. If the service is running on the mobile device, the proxy of the top-level service on the residential gateway is treated as the original service. Important is, however, that new proxy services encapsulating communication to the required services in the new location have to be generated accordingly on the mobile device.

3.5. Implementation Details

We use the Java-based OSGi component model for eHome services. OSGi provides a SOA-based runtime environment for services and applications. The mobile implementation is based on top eRCP, an OSGi implementation for embedded systems. Unfortunately, OSGi and, thus, eRCP do not support distribution of services over multiple gateways. Due to this, we have realized remote communication between the mobile device and the eHome gateway via WLAN based on JXTA, a language-independent P2P protocol. We implemented our own RMI-like communication over JXTA, called "SimpleRMI", to enable distributed service interaction over multiple gateways [1].

Furthermore, we have extended our configuration approach to support dynamic and distributed service composition and deployment [7]. A light-weight version of the eHome Base component for mobile devices is used for this purpose. The personal data is modeled and exchanged based on the user model markup language USERML. The services interpret the data according to the general user modeling ontology GUMO [6].

We evaluated the mobile device software on Dell Axim X51v PDAs, capable of WLAN. As Java virtual machine for mobile devices we used IBM's WebSphere Everyplace Micro Environment. The evaluation of the gateway side was done in combination with our existing eHome prototype. This prototype contains a 2D simulation environment, the eHomeSimulator, which can be executed on usual computers to simulate different smart environments [2]. We have tested several services such as the Wake-up, Music, or Personal Room Temperature etc.

4. Related Work

Mobile Gaia [5] is a middleware which enables adhoc personal active spaces. A user can integrate his mobile devices to a personal active space for realizing certain functionality. However, the authors do not describe how Mobile Gaia can be used for connecting mobile devices with usual smart environments, called active spaces in Gaia terminology. Furthermore, Mobile Gaia assumes that each mobile device runs applications either of coordinator or client mode. In our approach the a mobile device is used for different purposes. Besides running personal services, it can be also used to manage and release stored personal data.

The Aura project [8] aims at preserving continuity when a user moves between different environments. This is done by storing user task data on a global file server and by connecting the different environments to this server. In contrast to that, we let the users take along their data and even their services on a mobile device. This way, the user can control which parts of his personal data to release to a new environment.

Agents play an important role in the *MavHome Project* [9]. There are three main goals in this project: Maximizing living comfort, minimizing resource consumption, and maintaining safety and security of inhabitants. These goals are achieved by treating environments as intelligent agents. In contrast to that, we do not use agent technology but service composition based on an adaptive configuration process. Since we support different service realization patterns, we propose an approach based on a global view on the current state of an eHome system. Also, to our knowledge, inter-home mobility is not considered in the MavHome project.

Roduner et al. have analyzed the strengths and limits of using a mobile device as a universal interaction device in ubiquitous computing environments. They developed a system called AID for this purpose. Several tests have proved that persons using AID are faster solving exceptional tasks but slower solving every day tasks compared to executing these tasks on the appliances' own user interfaces. In our project we have also developed a prototype for mobile devices providing eHome users a unified user interface for interacting with personal and non-personal top-level services [1]. In contrast to AID, we additionally enable executing services and storing personal data on the mobile device for personalizing environments.

5. Summary and Outlook

In this paper we discussed different patterns of realizing eHome services. We evaluated these patterns and analyzed their applicability for mobility support and personalization. There is no single pattern that covers all scenarios that can occur in eHome systems. Therefore, we implemented a prototype which supports the execution of eHome services according to all discussed patterns. We could successfully show the applicability of our approach in a test environment consisting of a simulation environment [2] and several mobile devices.

We have also done some research on protecting the privacy of mobile users by use of anonymous credential systems. However, we could not discuss the results in this paper, a respective publication is pending.

The presented patterns need to be evaluated in a representative study in order to assess their applicability in real world scenarios. Therefore, we are looking for industrial partners providing a large-scale testbed.

References

[1] I. Armac and D. Evers. Client Side Personalization of Smart Environments. In *SAM 2008: Proc. of the* 1^{st} *Intl. Workshop on Software Architectures and Mobility at ICSE 2008*, pages 57–59. ACM, 2008.

[2] I. Armac and D. Retkowitz. Simulation of Smart Environments. In *Proceedings of the IEEE Intl. Conf. on Pervasive Services 2007 (ICPS'07)*, pages 257–266. IEEE Press, 2007.

[3] I. Armac and D. Rose. Privacy-Friendly User Modelling for Smart Environments. In *Proceedings of the The Fifth Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2008)* . ACM, 2008.

[4] L. Bass and J. Klein. Implications of a Single Mobile Computing Device. In *SAM '08: Proc. of the* 1^{st} *Intl. Workshop on Software Architectures and Mobility*, pages 51–52. ACM, 2008.

[5] S. Chetan, J. Al-Muhtadi, R. Campbell, and M. D. Mickunas. Mobile Gaia: A Middleware for Ad-hoc Pervasive Computing. In *IEEE Consumer Communications and Networking Conference*, pages 223 – 228, Las Vegas, Nevada, January 2005. IEEE Computer Society.

[6] D. Heckmann, T. Schwartz, B. Brandherm, M. Schmitz, and M. von Wilamowitz-Moellendorff. Gumo - the general user model ontology. In *Proceedings of the* 10^{th} *International Conference on User Modeling*, pages 428–432, Edinburgh, UK, 2005. LNAI 3538: Springer, Berlin Heidelberg.

[7] D. Retkowitz and M. Stegelmann. Dynamic Adaptability for Smart Environments. In R. Meier and S. Terzis, editors, *Distributed Applications and Interoperable Systems,* 8^{th} *IFIP WG 6.1 International Conference (DAIS 2008)*, volume 5053 of *LNCS*, pages 154–167. Springer, 2008.

[8] J. P. Sousa and D. Garlan. Aura: an Architectural Framework for User Mobility in Ubiquitous Computing Environments. In *WICSA 3: Proc. of the IFIP* 17^{th} *World Computer Congress - TC2 Stream / 3rd IEEE/IFIP Conference on Software Architecture*, pages 29–43. Kluwer, B.V., 2002.

[9] G. M. Youngblood, L. B. Holder, and D. J. Cook. Managing Adaptive Versatile Environments. In *PERCOM '05: Proc. of the* 3^{rd} *IEEE Intl. Conf. on Pervasive Computing and Communications*, pages 351–360. IEEE Computer Society, 2005.