Client Side Personalization of Smart Environments

Ibrahim Armaç  
Computer Science 3 (Software Engineering)  
RWTH Aachen University  
Ahornstr. 55, 52074 Aachen, Germany  
armac@i3.informatik.rwth-aachen.de

Daniel Evers  
Computer Science 3 (Software Engineering)  
RWTH Aachen University  
Ahornstr. 55, 52074 Aachen, Germany  
evers@i3.informatik.rwth-aachen.de

ABSTRACT
In this paper we will describe our approach for supporting users by personalizing the multiple environments they visit in their daily lives. In our approach, user preferences are stored on a handheld device which can assist the user in selecting, installing and parameterizing personal services in the environments.

Categories and Subject Descriptors
D.2.11 [Software Engineering]: Software Architectures—Domain-specific architectures; J [Computer Applications]: Miscellaneous

General Terms
Design

Keywords
Smart environments, personalization, mobility, OSGi, JXTA

1. INTRODUCTION
In this paper we will describe our approach for supporting mobile users in personalizing smart environments, in particular smart homes or, as we call them, eHomes. These are environments equipped with devices and appliances, which are usually controlled by a gateway. The gateway acts as a runtime environment for basic and integrating eHome services. Basic services, such as a lamp driver, do not use other services to realize their own functionality. In contrast, integrating services are composed of various basic and, optionally, other integrating services. Integrating services are called top-level services if they are not used by other services. We further declare top-level services as personal if their functionalities depend on user preferences.

One of the main challenges of developing context-aware eHome services, which strongly depend on user location, is the user’s mobility. There are two kinds of mobility important to us: in-home and inter-home mobility. In this paper, we will focus on the inter-home mobility: As Figure 1 denotes, users do not stay in a single environment all the time. Instead, they visit different environments in their daily lives, such as their workplace, hotels, friends’ homes etc. How can we support users in personalizing visited environments by using personal services, assuming that the environments provide the necessary infrastructure? Usually, users have to enter their preferences manually for each environment, which can be a tedious task. In this paper we will describe our solution for client-side personalization based on a personal mobile device (called handheld).

A simple example of a personal service which could be used in different environments is the Music-follows-Person service. Combining multimedia devices and person detection, this service lets a music stream associated with a specific user follow this user through the home. Other similar examples of services adapting environmental variables to user preferences are Light-follows-Person or Temperature-follows-Person.

2. PROBLEM DEFINITION
The following problems have to be addressed when personalizing multiple environments.

Consider that a user already uses the abovementioned personal services in his home. Now, if he wants to use the same services also in his holiday flat where he spends his weekends and these services are not installed there already, the user has to select, install, and parameterize them manually. However, to repeat these steps for each service in every environment is not feasible.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAM’08, May 10, 2008, Leipzig, Germany.

Copyright 2008 ACM 978-1-60558-022-7/08/05 ...$5.00.

57
Similarly, if the services are already running in the environment but the user comes there for the first time (e.g., an hotel room or a friend’s home), the services would have to be parameterized with the user’s preferences, again manually (the permission of the host is assumed). Because the environments do not share user preferences, the settings are stored separately in each environment. As a result, if the user changes his preferences, he has to propagate these changes to every environment manually. This will apparently discourage the user.

3. CLIENT SIDE PERSONALIZATION

The approach presented here is based on the idea that the user carries a handheld. This device can discover the presence of an eHome by detecting a corresponding wireless network and notifying the user about it. Furthermore, it can assist the user in performing tasks such as selecting, installing and parameterizing eHome services.

All communication with the eHome services as well as the eHome’s management facilities is based on well-defined interfaces. This ensures that the handheld knows all methods of the remote objects it communicates with, so that they can be used like local objects, similar to Java’s RMI.

3.1 Motivation for Using a Handheld

Today’s mobile devices are multifunctional. In our approach, we utilize the user’s handheld (e.g. a cell phone or PDA) for following purposes: (1) As a unified remote control for smart environments, (2) as a platform assisting the user in performing personalization tasks and (3) as a mobile storage device for user preferences.

Our work is focused mainly on the purposes two and three, i.e., using the handheld to assist the user in personalizing the visited environments. This is achieved by storing the user preferences on the handheld, to be available when needed; secondly by providing interfaces to the eHome’s management facilities, so that the handheld can select, trigger the installation of and parameterize services based on user preferences.

Furthermore, the handheld can be used for indoor localization as well as for the execution of top-level services.

3.2 Service Specifications and Interfaces

In order to provide automatic service selection by the handheld, the user has to take along the specifications of his personal services. In particular, a service’s specification contains the list of its provided functionalities, describing the service’s behavior. We describe functionalities by so-called “functional labels” [5]. E.g., the functionality of the Heating service can be described by the label Heating. If this service can be parametrized to regulate the temperature based on the current time, the specification can contain the label TimedHeating. It is also possible to order functional labels hierarchically, so that e.g. TimedHeating always includes Heating and extends this functionality.

To make services with the same functionalities interchangeable, we require that every label is associated with a unique interface, which has to be implemented by each service providing this functionality. This allows the usage of a service based on its specification, independent of its implementation. Thus, services with different names and implementations can be exchanged, as long as they provide the same functionalities.

3.3 Automating Service Selection, Configuration and Deployment

For access to the environment’s management facilities and to allow a semi-automatic configuration by the handheld, the environment has to provide amongst other components an Authenticator and a Service Manager (see Figure 2). The Authenticator is responsible for user authentication and access control. The Service Manager provides access to all service specifications and instances. The handheld can retrieve a list of personal services which the user is allowed to use. Depending on the user preferences, the handheld can request the Service Manager to install and/or instantiate selected services for the user. Every instance of a personal service is responsible for one user’s preferences.

The automatic parameterization of the selected (personal) services is the last necessary step in personalizing the environment. The parametrization can be automated if preferences corresponding to the selected services are stored on the handheld. Like functional labels, we assume that parameters also adhere to a consistent naming schema, where each parameter name is unambiguously defined.

3.4 Parametrization and Service Interfaces

Figure 3 gives an example of how the preferences on the handheld are stored. The user created preferences for temperature, light intensity and music. He has also stored the specifications of the personal services he would like to use in visited environments: Temperature-follows-Person, Light-follows-Person and Music-follows-Person. These services are running in his home environment along with other (non-personal) services, such as TV and Alarm. If the user now visits the environment on the right side, the two personal services running there would be instantiated and parameterized for the user.

Each personal service implements a method which returns a list of personal parameters. These imply getter and setter methods for each parameter; using them allows personalization of the service. For example, a heating service may have the parameter preferredTemperature. This implies the methods getPreferredTemperature() and setPreferredTemperature(). This way, the parametrization of personal services can be fully automated. Furthermore, user changes on preferences are automatically propagated to the corresponding services. As a result, the user is required to change his preferences only once. They are then transferred for every environment visited in the future.

Figure 2: Interaction handheld ↔ environment
environments to be connected together to migrate applications or services. Furthermore, in our work the users take along their preferences on a mobile device, allowing the personalization of visited environments without connection to a centralized infrastructure. This is required in Gaia for application mobility.

There exist also further research projects about context-aware smart environments, such as the MavHome Project [7], Nexus [3], the Adaptive House [2] etc. These projects focus on similar aspects, such as comfort enhancement and adaptation to the behavior of known users in single smart environments based on different technologies, such as agents or artificial intelligence. In none of these projects we found work supporting mobile users in client-based personalization of visited environments.

5. CONCLUSION AND OUTLOOK

In this paper we have discussed the client-side personalization of smart environments. Our approach is based on handhelds which assist the user in selecting, configuring and installing services. The user preferences are on a handheld and therefore available everywhere. The handheld can also assist the user in setting up personal services corresponding to his preferences and can act as a UI to the environments.

The presented solution provides starting points for further improvements. Some examples are standards for service specifications and corresponding interfaces (e.g. based on ontologies), service execution on handhelds, synchronization between the profile manager on the handheld and on the environment gateway, as well as conflict management for multi-user conflicts. Furthermore, security and privacy aspects have to be considered. We are working on negotiation-based authentication using anonymous credentials and role-based access control.

6. REFERENCES