Privacy-Friendly User Modelling for Smart Environments

Ibrahim Armac
Department of Computer Science 3
RWTH Aachen University
Ahornstr. 55, 52074 Aachen, Germany
armac@i3.informatik.rwth-aachen.de

Daniel Rose
Department of Computer Science 3
RWTH Aachen University
Ahornstr. 55, 52074 Aachen, Germany
rose@i3.informatik.rwth-aachen.de

ABSTRACT

In this paper we will describe how we model personal data in smart environments, which are user adaptive systems. Personal services in such smart environments can access user data by a consistent interface of our user modelling system. Considering inter-home mobility (i.e. users moving across multiple smart environments), the user model is flexible in the sense that it can dynamically deal with users entering or leaving a smart environment. Applying anonymous credentials and pseudonymity in combination with role-based access control, we can protect the privacy of users while protecting the environments against malicious users.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures*; K.8.m [**Personal Computing**]: Miscellaneous

General Terms

Design

Keywords

Smart environments, user model, personalization

1. INTRODUCTION

In this paper we will describe our approach for user modelling in smart environments, particularly smart homes, which we call *eHomes*. These are environments with devices such as sensors or appliances connected to a hardware platform, the residential gateway. On it runs a software platform, the service gateway, where we can run *eHome services*. We distinguish different types of services: *Basic* services act as drivers for devices, allowing us to abstract from the hardware and/or protocols used. *Integrating* services are composed from services, basic and/or integrating, delivering higher-order functions. Integrating services offering functionality which the user might be directly interested in (such as lighting, heating or music) are called *top-level* services. Lastly, top-level services which react on *user data* (we say also personal data) are called *personal* services.

MobiQuitous 2008, July 21 - 25, 2008 Dublin, Ireland

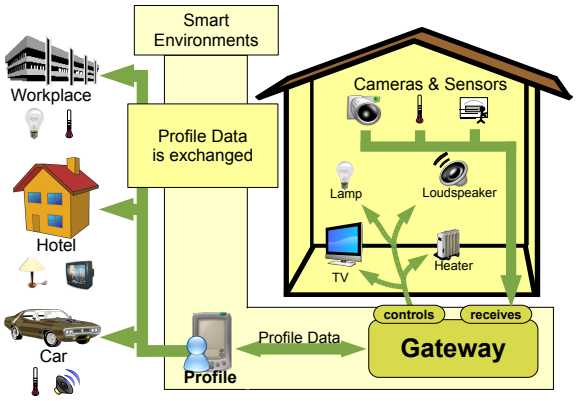

Figure 1: Inter-Home Mobility.

In particular, we are interested in users moving between multiple environments (which we call *inter-home mobility*), such as between their home, their office, or a visited hotel, as shown in Figure 1. We want to enable hassle-free access to these differing environments, while allowing users to keep their preferences for services across these multiple environments. These preferences include the services the user wishes to use (such as heating) as well as the settings for said services (such as the preferred temperature). Therefore the visited environment must have access to these preferences, possibly changing them based on user input.

Thus, we need a component for user modelling. Pohl defines a user model as: "a source of information, which contains assumptions about those aspects of a user that might be relevant for behavior of information adaption."[10] We therefore want a component in the eHome's service gateway holding user information and providing it to requesting personal services. Additionally, a similar component should exist in a user's mobile device, allowing them to take their data with them. Lastly, synchronisation between these two components becomes necessary.

Furthermore, we want to ensure the privacy and security of user data. We understand privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others."[14] Security is a much more loosely defined term; we understand it as all measures necessary to uphold the user's privacy, such as access controls, authentication, encryption, etc.

As the behaviour of services and thus the eHome relies on user action and data stored in the user model, we consider this a *user adaptive system*. This encompasses our middleware system, the

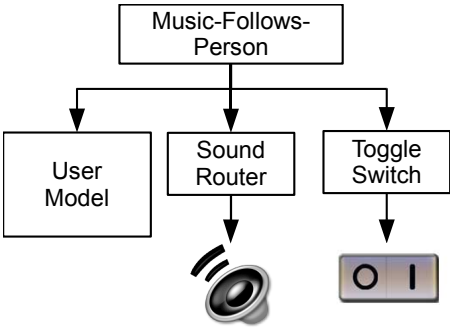

Figure 2: The Music-Follows-Person Service.

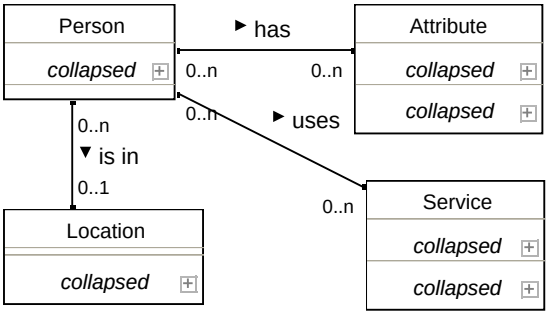

Figure 3: Person-related Part of the former eHome Model.

user modelling system, the data stored in the user model, and the services, all responding and adapting to changes in user data.

This paper is structured as follows. In the next section we will introduce a small scenario. Then we will detail the problems we were faced with our initial work plus the new features which we want to add for our user modelling system. This is followed by our solution for user modelling in eHomes. After that we will give a short overview on related work by other research groups. Lastly, we will close with a conclusion and outlook for further research.

2. SCENARIO

To help clarify the user modelling system and its interaction with the rest of an eHome system, we introduce an example of a personal service: The `Music-Follows-Person` service plays the user's favourite music in the currently occupied room. The basic set-up can be seen in Figure 2. The service accesses the user modelling system for information about the tracked person such as his current location as well as his preferred music. Then it uses a `Sound Router` service for playing the music at the user's location, while the user can turn the music on or off via a `Toggle Switch` service.

Our scenario consists of Bob visiting his friend David's house for the first time. Bob loves music, mostly Jazz, but also Reggae. He wants to use the Music-Follows-Person service, if possible, but at the same time would prefer to remain anonymous (to the service gateway) as far as possible.

3. PROBLEM DESCRIPTION

We have developed an extensive middleware tool suite for supporting the specification, configuration, and deployment of services, realized as OSGi bundles [12], in an eHome [9]. Part of this system is a large global data model holding information about the running services, the environment, and other contextual information. This data model is called the *eHome model*. In a first step, we will detail how user modelling was previously handled by our tool suite, followed by new features we want to implement with our new user modelling system.

In the past, no particular thought was given to user modelling in our system. In the beginning, there was no part of the eHome model explicitly for personal data. Each service could store arbitrary data into the global data model as attributes, which encapsulate an object. User data was therefore stored as attributes of individual services, as needed.

Later the eHome model was changed, adding an explicit person class (see Figure 3). Each person object, representing a *previously-known* user, could have attributes; this was used for storing any personal settings [8].

This meant that services had to do most of the user modelling themselves. As part of its own initialisation routine, each personal service had to ensure that needed person objects and attributes had been correctly initialised. This led to duplicitous code, complicating maintenance and the addition of new features. Using a separate system responsible for user modelling obviously greatly reduces this.

Additionally, our tool suite itself had no UI components specifically for user modelling. Therefore, as part of our user modelling system we should have UI components for viewing and manipulating user data in a consistent way.

The eHome model is completely accessible and modifiable by all services. Since we wanted support for the security and privacy of user data, we decided to take all user data out of the model. These aspects should then be covered by the user modelling system.

Lastly, our tool suite had little support for saving and restoring of data. Saving and restoring only the user data was not possible at all, something we consider essential. This should also be handled by our user modelling system.

4. OUR APPROACH

Since we are interested in the mobility of users and in particular providing their data to visited environments, we had to decide on how to enable this. We decided against having a central (Internet-based) repository for the data, or interconnecting visited eHomes and the user's "home" environment. These approaches require trusting the central repository (something many users are not liable to do) respectively telling the visited eHome where one is coming from (making anonymity difficult). These issues become more apparent if one considers that this can be sensitive data such as medical data.

Thus we focused on integrating mobile devices (such as a PDA) into our system. As was seen in Figure 1, we assume that each mobile user has such a mobile device. This device holds personal data and is used to gain access to an eHome. This allows the transfer of preferences from one eHome to another [1], which is then used by personal services to adapt their behaviour to the user's preferences. Additionally, using a mobile device allows use of its computing power and storage capacity as well as using the device to present a uniform UI irrespective of the visited environment. Last but not least, mobile devices could also be used for indoor localisation of the users.

The basic architecture of our core user modelling system can be seen in Figure 4. The most important parts are the `Authenticator`, the `Profile Manager`, and the `Session Manager`. Details of the figure will be described in the following sections. Additionally, we have a graphical front-end for entering, viewing, and modifying user data as well as for setting access controls.

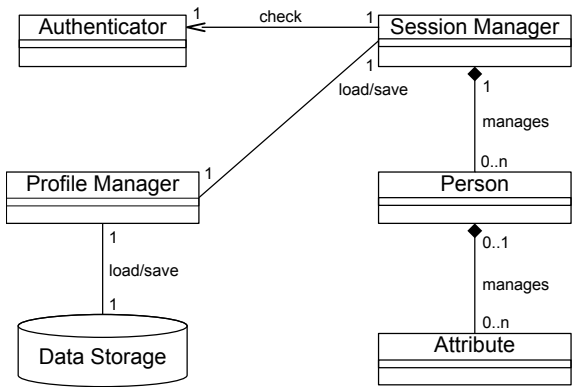

Figure 4: UM System Architecture.

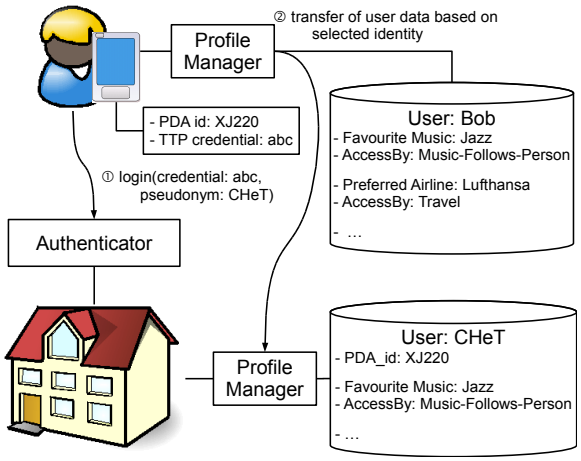

Figure 5: Login to an eHome via PDA.

The user modelling system is part of our protoype on the service gateway and has also been ported to the mobile devices. In the following sections, the user modelling system is described from the point-of-view of the service gateway, unless otherwise noted.

4.1 Authenticator

There exists one Authenticator in the system, which is used for two purposes: for user authentication and for access to services as well as user data.

To gain access to an eHome, the user has to authenticate to the visited environment using anonymous credentials (currently based on *idemix* [3]). These are zero-knowledge based certificates usually granted by a third party (alternatively also by the eHome itself) trusted by both the user and the eHome. Based on the credential shown, the user in granted a certain pre-defined role in the environment for enabling role-based access control. Pre-defined means that the eHome's administrator has defined for each role which services a user with said role may access. Having such a role, the user can access only services allowed for the role. Using anonymous credentials, the user does not have to be previously known by the system, nor is any administrative action necessary.

For example, Bob has a credential proving that he is a "friend" of the eHome's owner. The eHome can check the validity of the credential and therefore grant Bob access. The corresponding role might allow Bob to access the `Music-Follows-Person` service, but not the `Alarm-Configuration` service.

At login, the user can choose to be known by a pseudonym. We allow both *pseudonymous identification*, i.e. reusing the chosen pseudonym in future sessions (allowing the system to keep user data), and *anonymous identification*, i.e. using a new pseudonym each time (invalidating any user data after logging off) [11]. The latter choice enables us to achieve unlinkability of different sessions of a user in the same environment.

The mobile device might be storing a large amount of user data, not all of which is needed by the current environment. Because of data economy, it would be wise to give the eHome as little data as possible. One way in which we realise this is by allowing the user to choose an *identity*. This is a subset of the user's data, such as his preferred music, but not his favourite TV channel. Depending on the situation and visited environment, the user may have predefined identities such as "work," "home," "travel," or "gym." Only the data which is part of the selected identity will be made available by the mobile device to the environment. These features allow fine-grained user control what data is transferred to and known by the visited environment and control over the level of anonymity.

Secondly, the authenticator issues and checks credentials, which are given to personal services for access to the user modelling system. This is described in more detail in section 4.3.

Continuing our example, Bob logs into David's eHome pseudonymously, with the randomly chosen pseudonym of "CHeT." Note that at this point none of his personal data is yet known by the eHome. He chooses the identity "friends," which includes the attribute `Favourite Music`.

4.2 Profile Manager

The Profile Manager is mainly responsible for persistent storage of user data. It can save and restore data of individual users or the entire system. The data is encrypted with Triple-DES as proof-of-concept for protection of the saved data [13].

Additionally, communication about user data with the mobile devices is also handled by this component. After login, the Profile Manager of the user's mobile device and the visited environment synchronise: User data which is part of the selected identity is transferred from the mobile device to the service gateway in the eHome. This data is then available to the personal services. Before a user logs off, several steps are done. As a user may have changed his preferences during the session or services may have inferred new preferences based on the user's interaction behaviour, the Profile Managers synchronise again. Updated preferences are transferred to the mobile device. If the user has logged in using anonymous identification, all user data is then deleted on the service gateway. Thus, the eHome cannot link usage patterns of the same user in different sessions.

If user data is requested by a service, but is not found in the runtime data of a person, the Session Manager sends a request for said data to the Profile Manager (see Figure 4). The stored data is then searched for the requested information. If not found there, a request is made to the Profile Manager of the person's mobile device. If the requested data is available there, the Profile Manager of the mobile device (and optionally the user) can decide if the data should be transferred to the eHome. If the data was not found, the user can then be asked via the GUI described in section 4.4.

In our example, after logging into the system with his mobile device, some of Bob's preferences are transferred to the Profile Manager of the eHome. This is shown in the second part of Figure 5.

A new (pseudonymous) Person object is then created by the Session Manager along with his preferences as Attribute objects. Since his settings indicate that he wants the service `Music-Follows-`

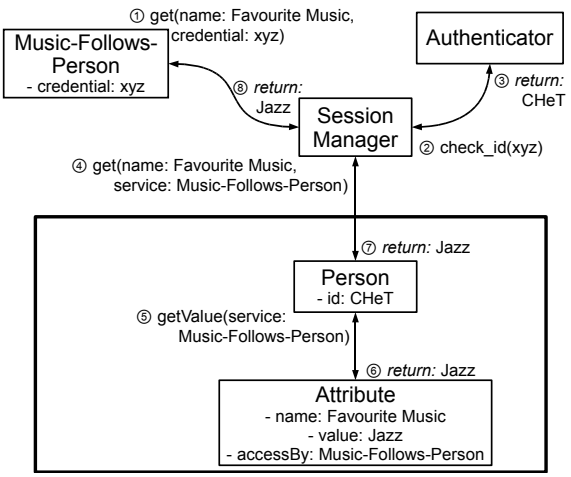

Figure 6: Service Requesting User Data.

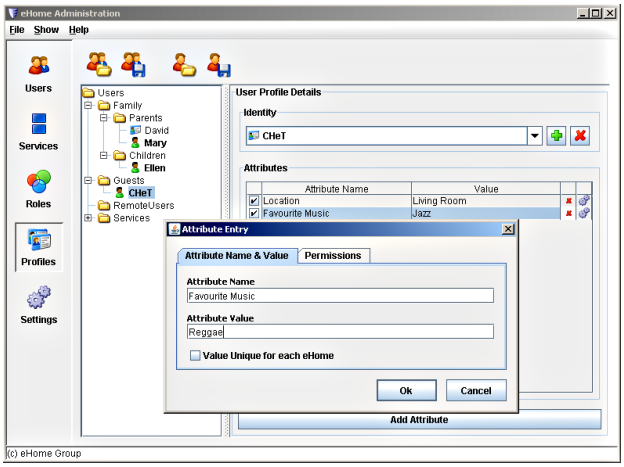

Figure 7: Part of our Graphical Front-End.

`Person`, a new instance of the service is started. The service gets a credential for access to the user modelling system, allowing access to some of Bob's user data such as depicted for the `Favourite Music` attribute.

4.3 Session Manager

The Session Manager is responsible for all runtime user data. All requests for and updates of user data by services are handled by this component. It is therefore the only part of the system visible to services. Also the creation and deletion of persons and their data is managed by the Session Manager.

As can be seen in Figure 4, the Session Manager creates a Person object for each (active) person in an eHome. The number of active persons is variable (`o..n`). The user data itself is then stored in Attribute objects (e.g. one for each preference), which encapsulate a generic object. This allows storing of arbitrary data. Besides the usual functions of getting and setting user data, a service can also register as listener for user data and therefore react to changes in said data (such as a changed setting).

As previously mentioned, the services need a credential for access to the user data. As part of this credential, we encode the person assigned to a personal service. Therefore, a service only knows that it is responsible for a person (getting access to the data via the credential), but it does not know the person's pseudonym or what other services have been deployed for that person. This way *anonymous service usage* can be realised.

As a further step of data economy, access to user data is done on a service-level basis. For each attribute, the user can set which services are allowed access to the data (indicated by the `accessBy` entry in Figure 5 and Figure 6). This gives very fine-grained control over access to his data. This technique is called *confidentiality by selective access* [11]. This way, we can also prevent users in the eHome gaining access to other users' data.

In our example, the `Music-Follows-Person` service needs to know the preferred music style of Bob (see Figure 6). With its credential, the service can ask the Session Manager for that data. The credential is decoded and validated by the Authenticator. The Session Manager then knows that the `Music-Follows-Person` service of CHeT (the pseudonym of Bob) is asking for that information. It can then search for the requested data and check if the service is allowed to know the data. Since this is the case, the service gets the requested data.

Combined, this greatly reduces linkability. Since the services do not know the person they are assigned to, they would have to compare the attributes to see if they are assigned to the same person. If the user has smartly chosen the access controls, allowing access to an attribute only if a service absolutely needs it, then the overlap of attributes shared by the services will be very small to nonexistent.

For this to work, the user has to trust that our system is working as advertised, since obviously the user modelling system has access to all of its data (see our future work, section 6). However, only part of the user's data may have been transferred to the eHome, due to identity management. Also, we are currently trusting the service gateway in protecting against a hacker directly reading the memory of the system.

To show why it is important to reduce linkability, consider the case of a smart hotel hosting a conference. If two attendees visit the sauna together, go to the bar and then watch a blue movie together, this could be used to blackmail these attendees. Without access control to user data and anonymous service usage, a malicious service could easily get this data. Having only access control, the three services for sauna, bar, and TV all would have to be malicious and share "their" data get the whole information about the users' service usage. If we would have only anonymous service usage, a single service could get all required data, but still would have to find out the real identities of the persons. With both, even with all three services being malicious, they cannot combine the data and disclose the identity of the users.

Another important function of the Session Manager is to manage location information of users. We have developed a person detector service which informs the Session Manager about user movement in the environment. When a user's location changes, the Session Manager informs the corresponding personal services, allowing them to react. The person detector service abstracts from the detecting technology, allowing us to use the many different sensor types being developed for location detection, such as Bluetooth, RFID, mobile devices tracking, etc.

4.4 Graphical Front-End

We developed a graphical front-end for users and their data as well as services. Role-based access control can be set via this front-end. Additionally, a dialog box for entering and manipulating individual user data and access controls was developed.

Parts of the GUI on the Gateway are exemplified in Figure 7. In the eHome, several persons are known: David, Mary, Ellen, and

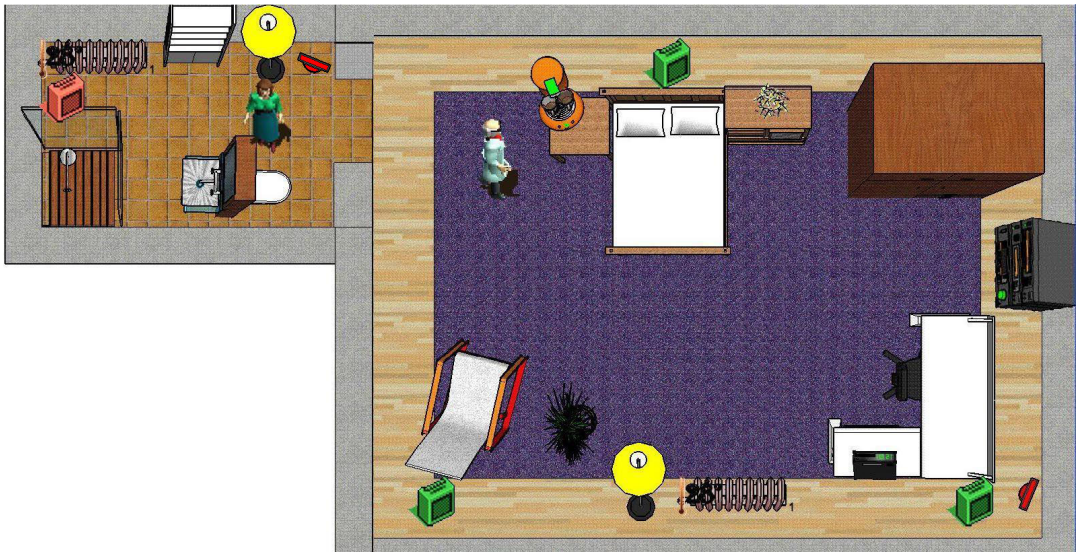

Figure 8: Screenshot of the eHomeSimulator.

Bob (under the pseudonym CHeT). We are currently viewing the profile of Bob.

Bob has a number of attributes, which can be edited or deleted by him. He is currently changing the attribute "Favourite Music" to reflect his new choice of "Reggae." In the second tab of the dialog box, he could change the access permissions of the services, such as to allow other services access to this data. This dialog box is also used whenever a service asks for an attribute with no known value.

We also ported the GUI to mobile devices (running Windows Mobile 5 or newer) and adjusted to the limited resolution available. The users can edit their preferences and identities alternatively by the mobile device or on a corresponding display in the eHome.

4.5 Evaluation

We evaluated the mobile device software on a PDA, a Dell Axim X51v, capable of Wireless LAN, in combination with our existing eHome prototype. This prototype contains a 2D simulation environment, the eHomeSimulator (see Figure 8), which can be executed on PCs to simulate different smart environments [2]. The implementation is based on Java on top of the Eclipse Embedded Rich Client Platform, which uses the OSGi framework for providing a plug-in based architecture. OSGi provides a SOA-based runtime environment for services and applications. The communication on top of the wireless network is based on JXTA, a language-independent P2P protocol. We implemented our own RMI-like communication over JXTA, called "SimpleRMI." We used IBM's WebSphere Everyplace Micro Environment as the Java Virtual Machine on the PDA.

For communication, we set up a WiFi network using a standard router. The eHome discovery is done by the JXTA framework. It searches the network when the wireless interface connects to an access point. The existence of a JXTA peer group named "eHome group" indicates the presence of an eHome gateway. After the mobile device joins this group, the user is notified that an eHome was found. He can then decide to authenticate or not.

We tested three personal services: After the authentication of the user using the PDA, the eHomeSimulator adapts the light intensity, the temperature, and the music in the current room to the user's preferences if they are part of the selected identity on the mobile device. Preference changes on the mobile device (e.g. temperature) cause the eHomeSimulator to adapt automatically to the new values in the corresponding room. We have also developed simple UIs on the mobile device. Thus, the remote control of the eHomeSimulator is possible.

Note that the location detection is done by the eHomeSimulator based on the coordination of the person figures in the simulator. This information is then abstracted by the person detector service and provided to the Session Manager, which passes this information to the personal services. In real environments, we could easily replace the detection technology. This also shows that the mobile device must not be carried all the time, rather being needed only if the mobile device is used to locate the person. Otherwise, it is only used for synchronisation of user data.

5. RELATED WORK

Research on user modelling has been ongoing since the late seventies. A great overview on the historical development as well as current topics is given by Kobsa in [6]. He identifies several future trends, such as mobile user models, including user models in smart appliances, and agent-based user modelling.

An example of an agent-based approach to user modelling is given by Lorenz in [7]. He defines four types of agents: sensory agents, modelling agents, controlling agents, and actuating agents. At each component, such as a PC, server, or mobile device, brokers are used in combining these agents and communicating with other brokers, thereby enabling distributed user modeling. However, while the specification has been presented, the actual implementation is still a work-in-progress.

Kay, Kummerfeld, and Carmichael have been working for a long time on user modelling. After implementing the `um` tool kit and the `Personis` user model server, they are now working on a version of Personis for resource-constrained devices, such as mobile devices, called `Personis-lite` [4]. They are using an accretion-resolution representation, gathering user data which is only processed when a request is made to the system. In comparison to our work, there is no support for anonymous service usage, meaning users have to be previously known to the system.

Lastly, as part of their work on interaction in smart environments, Hämmerle, Wimmer, Radig, and Beetz have developed a multi-

agent based platform called sHOME [5]. Sensor data is interpreted by agents, which forward semantic information to a central "brain." There it is combined with manually entered semantic information, preferences, and roles. Their main focus is on the interpretation of this sensor data: localizing and recognizing persons, where these persons are looking, locating the face and recognizing facial gestures, etc. This project does not focus on privacy issues, in contrast to our work.

6. CONCLUSION AND OUTLOOK

In this paper we have described our user modelling approach for smart environments. Because personal services depend strongly on personal data to be able to adapt their behaviour to user preferences, a user model is necessary for such environments.

The user model is divided into two parts: the Profile Manager and the Session Manager. The Profile Manager is responsible for the persistent (encrypted) storage of personal data on the gateway. Furthermore, it can synchronize with the Profile Manager on a user's mobile device. This is needed when a user authenticates himself to the environment or when services request personal data not part of the current identity. The Session Manager holds a runtime version of the user's data for each active person in the environment. It provides confidentiality by selective access for each personal attribute of a person.

We call our user model privacy-friendly for two reasons: First, we can use pseudonymous or anonymous identification, disconnecting a user's personal data from the user. Secondly, we combine this with an identity management system. Thus, a user is able to disclose only the parts of his identity which are necessary for the specific environment or situation.

Additionally, we have implemented a role-based access control system for protecting eHomes. This means that only users with the appropriate roles are able to use a service in the visited environment. As a result, the environments are protected against malicious users.

The presented solution provides starting points for further improvements. The most important, which we will work on in the future, is a conflict management strategy for multi-user conflicts. So far, conflicts are not detected when users with conflicting preferences meet in the same room. A possible solution would be adding a resolution strategy into the user modelling system, which could then resolve conflicts based on user priorities or through negotiation.

Also, we did not implement some common functionalities of many user modelling systems, such as stereotypes or inferential capabilities, because we did not need them for our purposes yet. However, they could be necessary in the future for personal services needing to infer new information about users based on already existing data.

Currently the wireless communication between the mobile devices and the service gateway in the environment is not secured. In the future it should however be encrypted in order to protect the transferred user data.

Lastly, the user currently has to trust that the system is working as advertised and has no back-doors or has been hacked. This could be reduced by issuing certificates (similar to those for web services, such as by VeriSign®) and by having internal parity checks.

7. REFERENCES

[1] Ibrahim Armac and Daniel Evers. Client Side Personalization of Smart Environments. In *Proceedings of the International Workshop on Software Architectures and Mobility at ICSE 2008 (SAM 2008)*. IEEE, 2008. (to appear).

[2] Ibrahim Armac and Daniel Retkowitz. Simulation of Smart Environments. In *Proc. of ICPS 2007*, pages 257–266. IEEE Press, 2007.

[3] Jan Camenisch and Els Van Herreweghen. Design and Implementation of the idemix Anonymous Credential System. In *Proc. of the 9th ACM Conference on Computer and Communications Security (CCS 2002)*, pages 21–30, New York, NY, USA, 2002. ACM Press.

[4] David J. Carmichael, Judy Kay, and Bob Kummerfeld. Consistent Modelling of Users, Devices and Sensors in a Ubiquitous Computing Environment. *User Modeling and User-Adapted Interaction*, 15(3-4):197–234, 2005.

[5] Simone Hämmerle, Matthias Wimmer, Bernd Radig, and Michael Beetz. Sensor-based Situated, Individualized, and Personalized Interaction in Smart Environments. In Armin B. Cremers, Rainer Manthey, Peter Martini, and Volker Steinhage, editors, *INFORMATIK 2005 - Informatik LIVE! Band 1, Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e.V.*, volume 67 of *LNI*, pages 261–265, Bonn, Germany, September 2005. GI.

[6] Alfred Kobsa. Generic User Modeling Systems. In Peter Brusilovsky, Alfred Kobsa, and Wolfgang Nejdl, editors, *The Adaptive Web: Methods and Strategies of Web Personalization (LNCS 4321)*, pages 136–154. Springer-Verlag GmbH, Berlin, 2007.

[7] Andreas Lorenz. A Specification for Agent-Based Distributed User Modelling in Ubiquitous Computing. In Peter Dolog and Julita Vassileva, editors, *DASUM 2005: Workshop on Decentralized, Agent Based, and Social Approaches to User Modelling, User Modeling 2005: 10th International Conference (Proceedings)*, July 2005.

[8] Ulrich Norbisrath, Ibrahim Armac, Daniel Retkowitz, and Priit Salumaa. Modeling eHome Systems. In Sotirios Terzis, editor, *Proc. of the 4th Intl. Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC 2006)*, pages 1–6, New York, NY, USA, 2006. ACM Press.

[9] Ulrich Norbisrath, Christof Mosler, and Ibrahim Armac. The eHomeConfigurator Tool Suite. In Robert Meersman and Zahir Tari and Pilar Herrero, editor, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Part II, 1st International Workshop on Pervasive Systems (PerSys 2006)*, number 4278 in LNCS, pages 1315–1324. Springer, 2006.

[10] Wolfgang Pohl. Logic-Based Representation and Reasoning for User Modeling Shell Systems. *User Modeling and User-Adapted Interaction*, 9(3):217–282, 1999.

[11] Jörg Schreck. *Security and Privacy in User Modeling*. PhD thesis, Universität – Gesamthochschule – Essen, July 2001.

[12] The OSGi Alliance. OSGi Service Platform Core Specification. `http://www.osgi.org/osgi_technology/download_specs.asp#Release4`, August 2005. Release 4.

[13] United States. National Institute of Standards and Technology. Data Encryption Standard (DES). FIPS pub 46-3, National Institute of Standards and Technology, Washington, DC, USA, 1999. For sale by the National Technical Information Service.

[14] Alan F. Westin. *Privacy and Freedom*. Atheneum, 1967.