Customization of Secured Ubiquitous Environments via Advanced Profile Management

Heinz-Josef Eikerling, Stephan Flake, and Robbie Schäfer

Abstract — **This paper describes concepts for the handling of profiling data for the customization of ubiquitous environments. The presented approach takes into account the secured data management and dissemination of profile data over different types of physical storages and is addressed by the work done in the IST FP6 project UBISEC [1]. A major issue in UBISEC is the transparent maintenance of a user's context depending on device, application and environmental settings as well as user preferences and capabilities. The advanced profile management architecture features security (authentication, authorization, privacy, data protection) especially with respect to mobile and dynamic environments. Employing such an architecture, users can be assured that even with a large amount of monitoring and background technology, their privacy is not at stake and that they may feel save and comfortable while acting in a mobile context.**

Index Terms — **Communication systems, computer network security, data management, mobile communication**

I. Introduction

Security in ubiquitous computing [2,3] is an ongoing topic of research where new challenges are imposed by the consideration of contemporary business areas and technologies originating from the integration of public wide area networks (e.g., cellular, Internet), private corporate, SOHO and home local area networks.

To make this profitable not only to the business stakeholders but also convenient to the end-users, requirements concerning an advanced customization infrastructure for context-aware and personalized authorization and authentication services in heterogeneous networks (in public or at home) have to be addressed. According to an ISTAG study [4], the lack of trust is one of the major obstacles for a broader proliferation of ambient and ubiquitous technologies. We therefore see the provision of security features w.r.t. the processed personalized information as a key requirement here.

Within the IST FP6 project UBISEC, the desired infrastructure is proposed to be composed of several reconfigurable building blocks, which can be easily compiled into ubiquitous applications. Though the system mainly relies on the use of smart card technology to keep identity-related information, the according data management is capable of being easily and securely integrated with other sources of profile information. The integrated management approach hence constitutes a means for providing advanced customization features (e.g., personalization of service or network access by considering location contexts) by taking into account privacy and aspects of protecting computing devices, the installed software components, and personal user data including user profiles. Note here that recent work on context based management of multiple user profiles [5] does not yet take security issues sufficiently into account.

The paper is structured as follows: subsequently we will briefly introduce the main principles to customization management comprised in the UBISEC approach, followed by the basic technologies that are used. Afterwards we elaborate on security and privacy issues as they occur in the various application domains addressed by the project and how they can be resolved by utilizing the UBISEC approach. Finally, we give some brief information, how the system is applied within a demonstration scenario established in the home domain.

II. Customization Management Principles

A. Customization and Profiling

Within our approach, *customization management* is driven by situation-dependent (context-aware) secure management and access control to entities via a set of domain profiles. According to this, the customization management interacts with three major configurable functional blocks (as shown in figure 1): authentication of users and devices acting on behalf of users, configuration of the environment including the participating devices and services, and service access which includes interactions with other aggregated services, e.g. for accounting and billing. These staged functional blocks are surrounded by additional services for providing context information and managing the profile sets and personalization information.

A *profile* is regarded as a repository of structured data that has an influence on other entities. The profile is the place where properties, features, and characteristics as well as present and past status are stored. Concerning the domains supported by profiles we consider user, device, service, application, network and context profiles. These domain profiles are

H.-J. Eikerling is with Siemens Business Services C-LAB, Fürstenallee 11, 33102 Paderborn, Germany (e-mail: heijo@c-lab.de).

R. Schäfer is with Paderborn University C-LAB, Fürstenallee 11, 33102 Paderborn, Germany (e-mail: robbie@c-lab.de).

S. Flake is with ORGA Systems enabling services GmbH, Am Hoppenhof 33, 33104 Paderborn, Germany (e-mail: sflake@orga-systems.com).

The work described herein is funded by the FP6-2002-IST-1 project UBISEC, contract number 506926.

assumed to be linked to each other, e.g. a user profiled by an according user profile maintains several devices which are characterized by the associated device profiles.

Each type of domain profile has to support a certain set of generic operations: lifecycle operations (*create*, *delete*, *replicate*), editing operations (*composing*, *linking*, *modifying, and moving*), retrieval (*lookup*) and advanced operations (*versioning*, *synching*, *inheritance*, *evolution*). Additionally, there are domain-specific operations.

It has to be also noted that profile information may reside on different types of storages reflecting the different levels of degradation concerning the secured data management. For example, device characteristics might be stored on a networked resource, whereas identity information as part of the user profile is retrieved from tamper-resistant smart cards.

B. *Architectural and Modeling Principles*

The above requirements have been turned into a data model and a software architecture for customization management based on profiles. Assuming that the profile data is dispersed over the different storage devices (termed *bearers*), an architecture featuring the distribution aspect has to be devised. In our approach this is addressed by a straight-forward concept for locating and addressing profile information based on *Uniform Resource Locators* (URIs) [6]. Below, the central component dispatching requests for performing operations on profile data to the respective bearers (networked service like http, ldap etc., database, smart card) will be described in more detail.

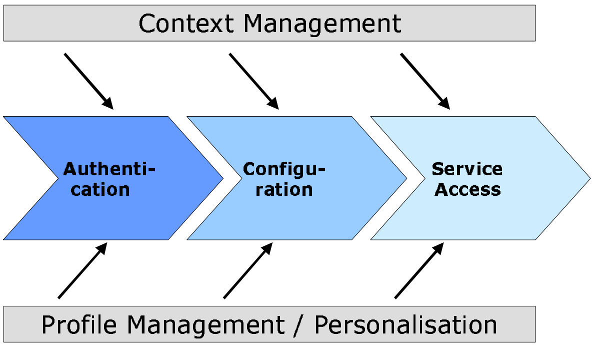


Fig. 1. Use of customization management in an ubiquitous scenario.

The transparent routing of requests for accessing and modifying profiles has an impact on the data model, i.e. meta-data has to come along with the profiled information which is intended to support the basic operations described above. As a result, the data model has to implement a container consisting of the domain-specific data payload and accompanying meta-data.

In order to have a homogeneous environment for controlling access to profile data, security related information is also included as part of the meta-data. For instance, when processing a request to lookup a certain profile this request has to be accompanied by the required credentials in order to retrieve the data from the bearer.

C. *System Architecture*

The core component of the Customization Management System (CMS) – as described in figure 2 – consists of the profile and personalization module which is composed of sub components handling the different types of bearers. The system hence constitutes a common interface for accessing profile information stored on the bearers. It contains one major sub component which is referred to as Profile Access Manager (PAM). The PAM dispatches a request according to the meta-information given by the location tag and the supplementary parameters. For instance, when receiving a request

```
smartcard://SC_ID?action=load&userid=id
```

the user profile of the user associated with the provided `id` is loaded from a local smart card (`SC`) and returned to the calling function or method.

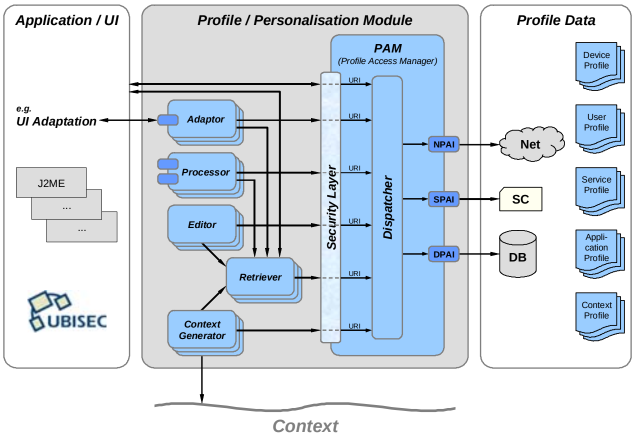


Fig. 2. CMS Functional System Architecture.

The bearers are accessed through access interfaces, which are controlled and called by the PAM. The PAM complies with an interface manager to which the different access interfaces are attached. Currently, the following types of access interfaces are supported:

- *Smart Card profile access interface* (SPAI):
  for load / store / update operations on a smart card
- *Network profile access interface* (NPAI):
  same as above for profile data residing on the net
- *Database profile access interface* (DPAI):
  same as above for profile data stored in a data base

The list of access interfaces depends on the supported location tags in the request URI format. For instance, the handling of profile data stored on some external storage like MMC or SD card would require an additional PAI (Profile Access Interface). The PAIs register with the PAM by providing exactly the information on the supported location tags, which are going to be processed by the respective PAI similar to the registration of a new context within a web application server.

D. *CMS Components*

According to figure 2 the architecture comprises three other major types of components accessing the PAM:

- *Adaptors***:** adaptors are used to adapt a resource according to the data stored in the profile. For instance, at the front-end the UI of an application may be adapted to the needs of the users which are captured by the according profile. Similar to the PAM, a manager for the set of adaptors is provided that permits to attach different adapters (e.g., UI adapter, media adapter,…) to the CMS.

- *Processors*: processors automatically adjust a profile data set. This can be done upon a request explicitly triggered by the user or by some rules, which need to be applied automatically. An example would consist of the handling of profile evolution through an according processor.
- *Editors*: editors are the front-end for the manual adjustment of profiles. Editors work on profile data independent of the profile bearer. Thus, they access the profile data only through the PAM (or through appropriate retrievers as noted below).

With respect to the implementation of the above components, they have to comply with a certain interface (on class level) and inheritance structure which is controlled by a manager. For example, generic operations for editing purposes, like opening and closing profile data files, are supported by a common editor interface. Quite naturally the PAM is not exclusively accessible via adaptors, processors, or editors but can also be directly accessed by UBISEC enabled applications.

E. Built-in Security

In our approach we apply different security technologies in order to secure the profile management. One such mechanism is access control to profile data and their bearers with a dedicated *access control specification language*, which permits to define access policies for profile data. Access policies are meta-data that have to be securely stored. Other CMS security mechanisms are encryption, hashing of profile payload data, and a security layer for tamper-resistance, authorization, and authentication:

1) Encrypted storage of sensitive profile data

Encryption/decryption of sensitive profile data can be applied at the profile element level, at the sub-tree level (element groups), or on the complete profile data. Encryption within a profile instance can be mixed, i.e., both element and sub-tree level encryption can be applied in a single profile instance. The profile schema defines which profile elements have to be encrypted when storing the data. The PAM keeps the encryption keys at a secret place, different from the location where the profile data is stored. Third-party applications can only access protected profile data if they know the decryption key themselves or if they contact the PAM. In the latter case, the applications have to provide credentials to proof that they are in fact allowed to access the requested profile data. Unprotected profile data can be accessed without providing credentials.

2) Hash values

The use of hash values (e.g., MD5) uniquely characterizes the profile data, so that tampering of profile data can be observed. The hash value has to be recomputed each time the profile data is changed. Hashing can be applied at the element level, at the sub-tree level, or on the complete profile data level.

3) Security layer

The PAM can be considered to act as a mediator between profile data holders (represented by the bearers) and the profile data accessors / mutators (represented by all components front-ending the PAM, e.g. adaptors, editors, and even applications). With respect to profile handling, end-to-end security between the bearers and the accessors and mutators has to be achieved. Therefore, a *security layer* is introduced to the PAM that addresses the following security features:

- *Tamper-resistant profile management.* If an accessor / mutator requests an action to be performed on a profile instance, the security layer provides a check-sum (e.g., MD5) which will be forwarded along with the request to the callee.
- *Authorization.* It must be ensured that only those requests are processed by the PAM that are sent by authorized profile accessors / mutators, i.e., the requester must have the privilege to access / modify the profile data under consideration. Two complementary schemes to ensure proper authorization are supported. Firstly, a PMI (Privilege Management Infrastructure) is applied. To establish a PMI in the context of the CMS, we have to additionally make use of a (potentially remote) Attribute Authority and a (local) Privilege Verifier. The Privilege Verifier communicates with the Attribute Authority to check for valid authorization. Secondly, authorization can be performed by other forms of credentials (PINs, passwords, pass phrases) that are provided together with requests. The security layer is then responsible to verify these credentials against the demanded information to access the profile data under consideration.
- *Authentication*. In order to be sure that the PAM, which is replying to requests, is the intended target entity, the PAM has to authenticate itself with requesting applications. Here, an enhanced public key infrastructure (ePKI) solution is applied which supports connected as well as disconnected modes of devices (w.r.t. a Certification Authority, similar to [7]).

III. Foundation Technologies

Various foundation technologies needed to realize the concepts described in the previous section have been identified. The use of the main foundation technologies, i.e., profiling, security, and service discovery, is outlined now.

A. Profiling

In order to manage profile based customizations, three concepts are of importance:

1. Means for structuring profile data (i.e., suitable profile formats) have to be established, ideally taking into account wide-spread notions and semantics.
2. Profile processing and evaluation methods have to be devised.
3. A runtime environment facilitating profile (or parts

thereof) exchange between the inner components of the CMS should be featured.

With regard to the UBISEC domain profiles, several existing formats have been evaluated. It was found that many domain-specific formats exist such as CC/PP [8], UA/Prof [9] for device profiles and PIDF [10] for user presence information, but that no format is supporting all targeted domain profiles (user-, device-, network-, application-, service- and context profiles). Common to the above mentioned formats is the use of XML and many of them rely on the Resource Description Format (RDF) [11] which allows expressing semantics of the described metadata. With regard to the CMS we decided to be compatible to existing formats whenever possible, even if internal formats for processing or storage might differ.

To allow sophisticated processing of profiles we rely on several artificial intelligence techniques such as Bayesian networks [12], fuzzy logic [13] and rule-based systems. A combination of these techniques allows both: profile evolution and learning as well as profile based decision making.

In order to implement the CMS architecture for distribution and retrieval of profiles we use the OSGi [14] framework, which facilitates a component based approach.

B. Secured Profile Access

To control access to profile data, the PAM checks credentials that are provided together with incoming requests. An appropriate access control specification language allows formulating *access control policies*. These are used to evaluate whether an access request is granted or denied. An access control policy is a set of rules of the form

```
(cred, profileId, accType, ctrlKind, scope)
```

where

- `cred` refers to a credential intended to proof the right of the requesting entity to access the desired profile data. The credential can, e.g., be a PIN uniquely representing a person, an application, or a group of people/applications.
- `profileId` refers to the desired profile instance to access,
- `accType` refers to the type of access and can be one of *create*, *delete*, *modify*, *read*, *write*, etc.,
- `ctrlKind` can be one of *grant* or *deny*, i.e., there are rules to explicitly grant or deny access,
- `scope` refers to the scope of the rule, e.g., a single element of the profile or an element together with its direct subelements or its complete sub-tree.

Several access control languages have already been defined and used in different domains. Standardized languages such as XACML and XACL are available as well as recent research efforts [16,17,18]. Independent of the employed access control language, the PAM has to *check access permissions* on incoming requests. (Note here that the requester might have been previously authenticated to enhance security.) For example, given a request of the form

```
req(profileID, cred, "read", profileElemList)
```

the PAM determines whether read access is granted as follows: If the profile determined by `profileID` is protected, check the credential `cred` for validity. If proven valid, the PAM takes the corresponding access policy and extracts all accessible (protected and unprotected) profile element data based upon the policy's access rules. Note here that for protected profile elements/groups corresponding credentials have to be provided as part of the `profileElemList`. From the extracted data, only the desired profile element data (i.e., data of the elements listed in `profileElemList`) are sent back to the requester.

C. Service Discovery and Access

Personalized access to digital media in heterogeneous environments through according localized services is a major scenario element within UBISEC. For convenient access to these service, dedicated service and network profile descriptions referenced through user preference profiles are provided. Employing this data, users can for example specify groups of services they are not interested in and those selected services will not be propagated for these specific users.

The according profile information can be used by means to ensure the secure handling of content over the entire life cycle – creation, manipulation, distribution and consumption – even in mobile environments [19] as well as architectural approaches to security in ubiquitous systems taking into account service-oriented architectures have been addressed here. With a specific emphasis on heterogeneous networks, the major challenges like handling service meta-information (formats, information propagation), discovery bridging, network topology management and protocol mapping / conversion have been examined. The security aspects for the broad range of networking and communication technologies (e.g., wireless Personal Area Networks like Bluetooth, cellular Wide Area Networks like UMTS/3G) on the different levels (e.g., RADIUS [20] on the service level) were also taken into account.

IV. Applying CMS to the Home Domain

A. Application Environment

For the validation of the CMS concepts described above mainly focusing on the use of device, context and user profiles, we concentrate on private home environments for two major reasons:

1. A closed environment, such as a home, is better controllable and measurable in the sense of proper evaluation of the concepts.
2. A home environment can easily accommodate the building blocks, depicted in the architecture above.

Home environments in ambient intelligence scenarios are typically very sensitive to the current user situation / context, while the situation is made up out of a variety of sensors ranging from simple ones (e.g. temperature, pressure, etc.) to audio and visual capturing devices. Together with the evaluation of user profiles, this monitoring creates a conflict between fundamental privacy aspects and the convenience achieved through home automation.

This conflict cannot be easily resolved, but a profiling approach that allows setting personal security preferences may help to assure the user that her / his privacy will be respected. In this sense, the context itself can influence the security level, as proposed in [21], which shows how different contexts can influence the type and quality of audio and visual data that may be transmitted in different situations, and thereby increasing privacy. For ensuring unobtrusive ambient security with respect to the user's identity, contactless smart cards as security tokens in conjunction with biometrics are employed. By that, they are to be used for authentication and access. These cards provide a most simple user interface for security relevant operations as accessing a restricted area e.g. by waving the card near a reader. Therefore the technology might be quite well adopted by a large group of home users, ranging from children to elderly people, as it does not require learning complex technology and there is also no need to memorize passwords[1].

However, because of the memory limitations of smart cards we foresee that personal profiles may be distributed also to home databases and even to networked locations that may be not in the home at all and need special protection.

B. *CMS Usage*

Within the current setup, the CMS is for the authorization and authentication of users to several services which for example ranges from entering the home (locking and unlocking the door) to online payment for specific services.

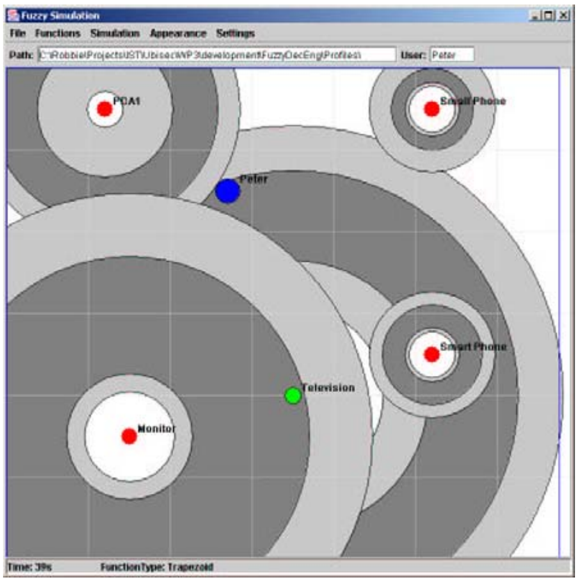
The editing components of the CMS are used to grant access rights to specific groups of persons for some applications. These permissions are stored in the user profiles and evaluated once an application is going to be used. Quite simply, it is possible to specify that children below a certain age are not allowed to watch TV in the evening. This shows also the context-dependant profile processing, as the current time is used to determine whether accessing the application is granted or not. Another context is given by the location, for example when moving to the office space of the father, access rights might be further restricted for the children.

Apart from managing the rights, the CMS controls automatic adaptation of services. This requires usage of almost all major components in following order:

1. Getting the context
2. Retrieving user and device profiles
3. Processing the profiles
4. Adapting the services

Now the following concrete scenario is considered: a user is in a room when an important message arrives which should be displayed on the most suitable device in the room. There are several devices scattered in the room like a TV-set, a flat screen, a PDA mounted in a cradle or a smart phone, each capable of displaying messages.

For the first step, the context retrieval component determines the location of each traceable device and the user. Thereafter, the profile retrieval component tries to obtain the device profiles of each active output device and the user profile. The profile processor now evaluates each profile but only needs certain properties. In this case, only the screen properties of the devices are relevant and the user's visual capabilities are of interest. Depending on the screen size, membership functions can be defined that provide information at which distance a display can be optimally viewed. We relied on trapezoid membership functions as there is usually a wide range where the display can be watched without problems. Since individuals have different sensing (viewing) capabilities, the results of the de-fuzzification (evaluation of the membership functions) are applied with an individual factor. Only devices which provide optimal solutions for the current situation are thus selected, and the information is passed to an adaptor which is responsible to trigger the selected output devices. In figure 3 we show a visualization of the simulated scenario. The simulation does the proper context calculation but abstracts from real positioning.



Path: C:\Robbie\Projects\IST\Ubisec\WP3\development\FuzzyDecEng\Profiles\

User: Peter

Time: 39s FunctionType: Trapezoid

Fig. 3. Handling location information for device assignment.

C. *Validation Approach*

There are many more kinds of usages for the CMS than can be described here. Final trials and validation based on the prototypes will be continued at the AC-LAB operated by Paderborn University and Siemens in Paderborn (Germany). The AC-LAB is set up as an artificial home with some separators to indicate different rooms and provides for example, multimodal and multi-device user interfaces that adapt to specific situations. This allows more intuitive, comfortable and easily traceable interaction. Different positioning and surveillance techniques e.g. based on W-LAN or on networked cameras provide means for increased personal security and context-based decisions, while contactless authentication with Java

[1] Passwords are still an option for even higher security which might be relevant for managing purposes, e.g. a family member with administration rights who configures the environment for other persons that don't want to bother with technical details.

Cards can be used for secure and easy access of rooms or services together with profile based customization.

For testing the CMS concepts, we rely on a set of fine grained scenarios and match the results provided through the system with the user's expectations. According to the above scenario, certain aspects can be effectively measured and matched with the subjective expectations of a user concerning latency for implementing a context change. For instance, we have evaluated the profile-driven redirection of streamed content (MPEG-2) to a device matching the user's preferences. The adaptation of the user's service environment (i.e., the target display) can be performed within a fraction (less than 10%) of the overall time for the redirection which includes the time for appropriate buffering. In general, the definition of metrics to measure the differences between real and expected behavior is currently done.

D. Discussion: Mobility Support through Customization

A major benefit of using profiled data and applying the concepts described in this paper are related to the management of the different types of mobility arising in wireless and mobile environments. Notice, that this is considered to be of high importance for future market evolution of mobile networks [23]. With regard to the notion of mobility in 3G wireless and convergence networks we focused ourselves on the *nomadic* use of such networks as opposed to the *seamless* mobility which requires consideration of all network layers [20]. According to this focus on nomadic mobility aspects, the following listing summarizes our findings:

- *User mobility*: we support the mobility of users (changing location plus potentially changing the device) by keeping track of the user's context which is handled as a specific type of context information.
- *Access device mobility*: the user may change the type or the particular instance of the device for accessing a service. In the application scenario a context change results in a different device assignment.
- *Service mobility*: within our design, we can model a service follow-up (e.g., routing streamed data according to context and other profile data, i.e. device characteristics).

V. Conclusion

We gave a succinct overview of the UBISEC customization management system (CMS) and discussed the results of validation within an application scenario. Since security is of high importance with regard to personalized information, the CMS approaches customization management by means of profiling while assuring privacy and secure access to profile information being placed on different sources.

The concepts and consequences / tentative results discussed in the previous sections indicate that user acceptance increases through ease of use and a trustworthy environment which is going to be backed by complementary trial scenarios built into the evaluation environments. In order to draw conclusions of the real user acceptance and benefits, further user studies are foreseen to be carried out.

References

[1] UBISEC website. Available at http://www.ubisec.org.

[2] Workshop on Security in Ubiquitous Computing, 2002. Available at http://www.teco.edu/~philip/ubicomp2002ws/index.htm 2002

[3] Second Workshop on Security in Ubiquitous Computing, 2003. Available at http://www.vs.inf.ethz.ch/events/ubicomp2003sec/

[4] IST Advisory Group. Ambient Intelligence - from vision to reality. *Draft consolidated report, 2003*.

[5] Salem, B. and Rauterberg. M.: Multiple User Profile Merging (MUPE): Key Challenges for Environment Awareness. *Proceedings of the Second European Symposium on Ambient Intelligence, EUSAI 2004. (LNCS 3295)*

[6] Dürst, M.J.: Internationalized Resource Identifiers: From Specification to Testing. *19th International Unicode Conference, September 2001, San Jose, CA.*

[7] Almenárez, F; Marín, A.; Campo, C. and García-Rubio, C.: A Pervasive Trust Management Model for Dynamic Open Environments. *First Workshop on Pervasive Security and Trust (PSPT 2004), Boston, MA, USA, 2004*.

[8] Franklin, R. et al.: Composite Capability/Preference Profiles (CC/PP): Structure. *W3C Working Draft, 2000.*

[9] Wireless Application Protocol: UA/Prof - User Agent Profile Specification. *WAP Forum, 2001*.

[10] Sugano, H. et al.: Presence Information Data Format (PIDF). *Network Working Group, IETF, Internet Draft, 2003.*

[11] Lassila, O. and Swick, R.: Resource Description Framework, (RDF) Model and Syntax Specification *W3C Recommendation, 1999.*

[12] Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, *Morgan Kaufmann, San Mateo, CA, 1988.*

[13] Zadeh, L.A.: Fuzzy Sets, *Information and Control, pp 338-353, 1965*

[14] OSGi Alliance. Available at http://www.osgi.org

[15] Micali, S.: NOVOMODO. Scalable Certificate Validation and Simplified PKI Management*. 1st Annual PKI Research Workshop, pp. 15-25, 2002*.

[16] eXtensible Access Control Markup Language (XACML). Available at: http://www.oasis-open/committees/xacml

[17] XML Access Control.
Available at: http://www.trl.ibm.com/projects/xml/xacl

[18] I. Fundulaki and M. Marx. Specifying Access Control Policies for XML Documents with XPath. In *SACMAT 2004, Yorktown Heights, New York, USA, June 2004.*

[19] Zheng, Y.: Mobile Digital Rights Management. *Seminar on Network Security, 2001. Helsinki University of Technology*.

[20] Rigney, C. et al.: Remote Authentication Dial In User Services (RADIUS). *Network Working Group, IETF, RFC 2865*.

[21] Neustaedter C. and Greenberg S.: The Design of a Context-Aware Home Media Space for Balancing Privacy and Awareness. *Proceedings of Ubicomp 2003.*

[22] B. Rao, L. Minakakis, Evolution of mobile location-based services, Communications of the ACM, pp. 61-65, Vol. 46, Dec. 2003.

[23] Y.-B. Lin and I. Chlamtac, "Wireless and Mobile Network Architecture", John Wiley & Sons, Inc., 2001.