The Personalization of Mobile Services

Ivar Jørstad, Do van Thanh and Schahram Dustdar

***Abstract*—This paper presents an analysis of the requirements for service personalization and proposes a generic service architecture that supports personalization. It starts with a study of relevant personalization works and a discussion on the importance of personalization on services. An information space for service personalization is elaborated. A definition of personalization is given, and based on this definition the components of a generic mobile service are identified. A summary of the requirements for each of these components is given. Two models for realizing personalization of mobile services are presented. Last, two case studies on personalized web browsing are presented to further highlight the complexity of personalization of generic mobile services, and to motivate for future work.**

***Index Terms*—customization, mobile services, personalization, user profiles**

I. INTRODUCTION

PERSONALIZATION of services is to adapt services to fit the needs and preferences of a user or a group of users. Personalization is important in today's service-oriented society, and has proven to be crucial for the acceptance of services provided by the Internet and mobile telecommunication networks. In [1], taxonomy of motivation for personalization is given, and two important categories of personalization are given: *personalization to facilitate work* and *personalization to accommodate social requirements*. In the first category, services are adapted to increase the efficiency, e.g. to minimize the time spent on repetitive and similar work tasks. The adaptation can aim at accommodating at physical differences of the users like weak sightedness, disabilities, etc. In the second category, services are adapted to enhance the social experience. For example, youngsters, by changing the appearance and behavior of a cellular phone (ring tone, logos etc.) want to express their personality.

To enable service developers and providers of both the Internet and mobile telecommunication networks to support personalization, adequate middleware and service platforms must be present. They act as a fundament and catalyst for increasing the number of personalized services. This paper investigates the requirements that must be fulfilled by such middleware and platforms in order to support fully personalized mobile services. In Section II the activities on personalization are discussed. In Section III, a definition of personalization is given, and the general process of personalization is illustrated. In order to further investigate the specific requirements of personalization for mobile services, this paper identifies the information space that contains the elements that are needed to allow personalization of services. In Section IV the components of a personalized mobile service are discussed. Two alternative models to realize personalization of mobile services, are presented in Section V. Section VI provides some illustrative case studies of how to enable personalization of a typical service, applying the two mentioned models.

Manuscript received January 10, 2005.

I. Jørstad is with the Norwegian University of Science and Technology (e-mail: ivar@ ongx.org).

T. v. Do is with Telenor R&D,Norway (e-mail: thanh-van.do@telenor.com).

S. Dustdar is with Vienna University of Technology, Austria (e-mail: dustdar@infosys.tuwien.ac.at)

II. RELATED WORK

There are a lot of works on personalization, like [1] that gives an overview of personalization of mobile services. However, most considerable in terms of standardization are the GPP's VHE and the W3C's CC/PP

A. 3G and Virtual Home Environment (VHE)

In the 3rd Generation Partnership Project (3GPP) specification [3], the VHE is defined as:

"A concept for personal service environment portability across network boundaries and between terminals."

A personal service environment (PSE) is defined as:

"[a PSE] contains personalized information defining how subscribed services are provided and presented towards the user. Each subscriber of the Home Environment has her own Personal Service Environment. The Personal Service Environment is defined in terms of one or more ***User Profiles****."*

Thus, the VHE's goal is to support mobile users with the same service environment no matter what network the user is connected to, and regardless of the user being on the road or at home. The basic idea in VHE is that a *foreign network* emulates

the behavior of the *home network*. VHE is, however, a concept and not a technology. Instead, it relies on other concepts and technologies to implement the functionalities needed to achieve its goals, e.g. the Mobile application Execution Environment (MExE) for the application runtime environment in the mobile equipment (ME) of the terminal (which in itself is dependent on other technologies like Java 2 Micro Edition), SIM Application Toolkit (SAT) for applications running on the Subscriber Identity Module (SIM), Parlay/Open Service Access (OSA) to enable exchange of functionalities between mobile telecommunication networks and Internet based services, etc. The current logical model for distribution of user profile data in VHE is shown in Fig. 1. Although the first high level user requirement in the VHE specification states that it should be possible for a user to personalize services, it is not clear what "personalized services" actually means.

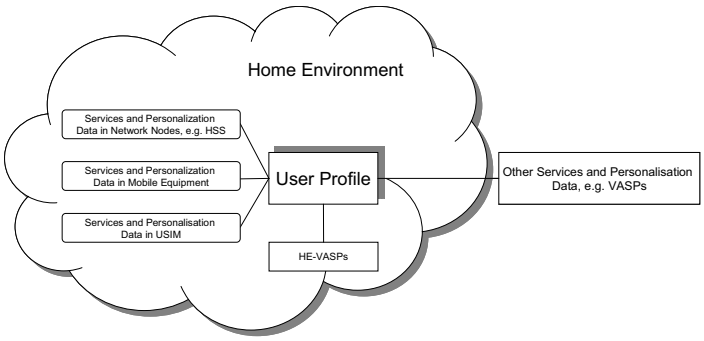

Fig. 1. Logical model of the user profile data distribution in VHE

VHE is a standard to be supported by 3G networks and is thus developed from a mobile telecommunication point of view. In the VHE specifications, services are treated generally and three types of services are identified:

- standardized services
- operator specific services
- other services

These categories are high level, and in order to be able to develop a service platform that allows personalization, it is beneficial to perform a more detailed study of what a personalized service is.

In the EU IST VESPER project [4], an architecture based on VHE is proposed in order to validate the VHE concept. The paper also provides a good introduction to VHE. Several entities that are required in a VHE implementation are identified throughout the paper. The proposed architecture defines a set of Service Components which offer a set of functionalities to support the VHE. The VESPER project demonstrates the use of VHE capabilities by using 3 applications [5]. Similar to the discussion in the rest of this paper, VESPER focuses on ubiquitous access to the personalized applications. An API enabling personalization of third parties service is identified. However, there is no solution to enable personalization of generic mobile services, especially the existing popular services like email, web surfing, mobile commerce, etc.

In [6], a VHE architecture for advanced value-added service provision in 3rd generation mobile communication networks is proposed. This paper focuses specifically on the execution environments that must be present on various terminals to support similar services, format for representing terminal capabilities, terminal capability negotiation and how to handle user profiles, among other topics. However, it is difficult to find treatment of how the particular composition of mobile services, and the fact that they should be ubiquitously available, dictates the requirements to the personalization enabling technology.

Another paper [7] discusses enabling technologies for personalizing mobile services and proposes an architecture for a Personal Service Environment (PSE). The paper focuses on the process of personalization, i.e., how to handle profile management, how to handle service discovery and how to adapt services. The paper does not cover what the requirements are necessary to enable personalization of mobile services, and it does not consider the composition of mobile services and how this relates to the feasibility for personalization.

Generally, current approaches to personalization do not study the architecture of a mobile service, i.e., what is its composition and what are the components. We believe that the service architecture has decisive impacts on the personalization process. Also, it is not clear how existing services can be personalized by the approaches presented earlier and what kind of services can be personalized.

B. Standardisation Efforts (OMA, CC/PP etc.)

Composite Capabilities/Preferences Profile (CC/PP) is a system for describing device capabilities and user preferences, specified by the World Wide Web Consortium (W3C). The CC/PP Working Group at W3C is closed, and the work is continued in the Device Independence WG. As it name implies, CC/PP has a clear focus on *devices*, similar to earlier work with personalization. UAProf by Open Mobile Alliance (OMA) implements CC/PP, allowing proxies to transform content for mobile use. Java Specification Request (JSR 188) CC/PP Processing defines a set of Java API for processing CC/PP and UAProf documents. A final release (1.0) of this specification was released in September 2003. According to the Java Community Process (JCP) it will enable interoperability between web servers and access mechanisms and facilitate development of device independent web applications.

In summary, CC/PP will enable device independence of mobile services, thus achieving some of the goals of personalization (similar services accessed from different devices). However, as it will be evident throughout the rest of this paper, device

independence is only one of the obstacles towards personalizing mobile services.

III. Personalization of Mobile Services

Various, quite different definitions of personalization can easily be found in e.g. [8] and [9]. However, throughout this paper the definition from [1] is used:

"Personalization of a service is the ability to allow a user U *to adapt, or produce, a* service A *to fit* user U's *particular needs, and that after such personalization, all subsequent service rendering by* service A *towards* user U *is changed accordingly."*

Although the definition states that personalization is done by the user, most of the tasks might still be done by *systems*. In fact, it only stresses that the user should the one in charge and initiating the personalization process in the first place. The general requirements that can be deduced from the definition are:

1. It should be possible to personalize services developed and offered by different service provider.
2. It should be possible for a user to spawn new services (i.e., Personalized Service Composition [1])
3. All components that have impact on the personalization process and features should be always available (i.e. ubiquity and Service Continuity [10][11])

These requirements raise three major questions:

1. What information is necessary to personalize a service (Identified at the design time)?
2. How is this information collected?
3. How is this information applied in the service rendering process?

To fully answer the first question, each service that should be personalized must be taken into consideration. However, it is beneficial to consider a generic mobile service first since it will help building a generic service platform and middleware for personalization. This question is further treated in Section IV.

Question 2 depends on an architecture and platform allowing certain information to be collected/captured and persistently stored, and question 3 depends on an architecture and platform allowing the same information to be retrieved and applied to a service. Both these questions, together with the third requirement from our definition of personalization, thus raise another significant question: Where should the collected information be stored? This question is not considered further in this paper.

Personalization relies on information about the user, one or several devices and the services. The resulting information space for personalization is depicted in Fig. 2. It consists of the User-service relationship, User-device relationship and Device-service relationship. These relationships have to be stored

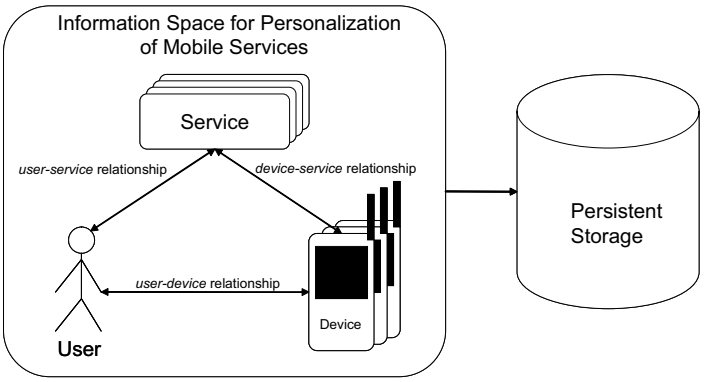

Fig. 2. The information space of service personalization

persistently in order to be restored and used in the personalization process.

A. User-service relationship

This relationship is characterized by information about the history of a user with respect to a specific service. This means, it covers all documents produced and consumed by the user through this service, as well as user-defined *behavior* of the service. Except for personalization features that are related to the *presentation* of a service, all other service specific personalization information is covered by this relationship, i.e., all *functional requirements* of a service are realized through this relationship.

B. User-device relationship

This relationship consists of information about which devices a particular user has at disposition, and particular adaptations of this device towards that specific user. This could for example be general settings with audio/video settings of the device and similar. These are settings that affect *all* services rendered towards the specific user through the specific device.

C. Device-service relationship

This relationship consists of information about how a specific device should behave towards a specific service and vice versa. This relationship is decided through the *user-device* relationship and exists only in the context of such relationships. It covers for example the look-and-feel of services, because this is both dependent on the particular service, but even more on the particular device and its characteristics.

CC/PP discussed in Section II, is particularly relevant to the *user-device* and the *device-service* relationship. If service platforms continue to support this standard, and user

requirements of look-and-feel can be easily negotiated between the user and the service platform, this part of the information space is covered well. Also, as seen earlier, terminal/device independence has been subject to a lot of work by various initiatives.

However, the *user-service* relationship, which includes most of the information that enables personalization, is not yet covered by any standard.

A general process for personalization can now be summarized by the following steps:

1. Collect information about the *user-service, device-service* and *user-device* relations
2. Persistently store this information
3. Use service
    a. Identify the user
    b. Identify the device
    c. Look up information about the relevant relationship for this user
    d. Render personalized version of service according to information about the *user-service, device-service* and *user-device* relations

Step one in the personalization process can initially be divided into two categories, namely *explicit* and *implicit*. With explicit personalization, the user controls the first step, whereas with implicit personalization the system controls the first step of the process.

Explicit personalization might be used for gathering information for customizing the look and feel and behavior of a service, while the other could be for highlighting the most used functions in a service, or as used in the WWW, for personalized marketing purposes. For the first type, the user might be given the opportunity to specify details, whereas the other could be based on mining logs from previous service usage history.

IV. Personalization Components in a Mobile Service

Having outlined a general process for personalization, it is now time to model a generic mobile service, i.e. to identify the components that a generic mobile service must have, in order to enable personalization.

The most central component of a mobile service is the *Service Logic* component, which in reality consists of one or more executable files. The execution state of a service is captured by internal parameters. If the service shall be mobile, i.e. available to a mobile user on any network and any terminal, it is better to gather these state parameter in a component called *Service data*. To realize personalization of a mobile service, however, this model is not sufficient.

Quite often, services are commonly used to either produce or consume some kind of content, e.g. documents. These are an important part of the service, and to personalize a mobile service is also to enable the service logic to access the right content at anytime. It is thus important to introduce an additional component called *Service Content* that contains meta data necessary for usage of content. For example, the personalized content of an Internet browser consists of bookmarks, history, current visited page (the URL in the address field), various preference settings regarding look and feel of web pages, cookies, saved passwords, etc. Currently, the Service content is quite fragmented and stored in several places and personalization is quite difficult. For some browsers and operating systems, bookmarks are stored in regular files in the operating system's file system, the same goes for the history of visited pages. Information about the look and feel can be stored in specialized preference files or in MS Windows it could be stored in the Registry.

It is worth noticing that after the identification of the information needed to apply personalization, the major challenge is to find where all this information is stored and it might prove even harder to retrieve it (proprietary document formats etc.).

To personalize functionalities, look and feel and the service portfolio etc. it is necessary to have a *Service Profile*. The first challenge with *Service Profile* is the format. It must be ensured that all personalized service can understand the format that is used to describe the service profiles. There are yet no standard for representing these, because there is no general agreement on what these profiles should contain. The second challenge with *Service Profile* is the storage location. It must be stored in a location where it is accessible from any service logic component that should be personalized. This leads to the third challenge, which is the ownership of the profiles. If users have problems trusting service providers, users should be the owners of the profiles and they should be stored in a way that ensures and protects their privacy.

The resulting architecture of a generic mobile service is thus as displayed in Fig. 3 and the components that must be considered for enabling personalization are further discussed below. Further discussions around the architecture of generic mobile services can be found in [10] and [11].

V. Two Models for Personalization

In this section, two models for personalization will be presented. The models are based on the components of a mobile service as described earlier in this paper.

The first model allows personalization of services by requiring the services to implement standardized interfaces towards the components that are used to realize personalization. The second

model allows existing services to be personalized, although as will be seen, some restrictions will apply.

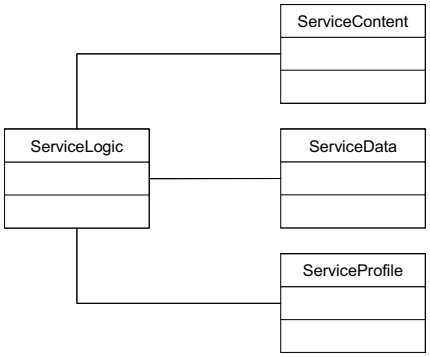

Fig. 3. Components that must be considered when personalizing mobile services

A. Model 1

As shown in Fig. 4 in this model for personalization of services, standardized interfaces towards the components are needed. SI_{a-c} are interfaces are the standardized interfaces towards Service Content, Service Data and Service Profile, respectively. ServiceLogic1 is the executable logic (application) running in one location (Location1). As the user moves to another location (Location2), he resumes operation of the service with a new service logic component (ServiceLogic2). This service logic also implements interfaces SI_{a-c}, and thus can access the information that is needed to continue the provisioning of a personalized service. The information itself (service data, content and profile) are stored in a shared location, which could be a data haven somewhere in the Internet, or in a Personal Service Environment (PSE) as defined in the VHE specifications.

Opportunities and Challenges with Model 1

This approach requires designers and implementers of new services to follow the specifications of the standardized interfaces. Some of this work (regard parts of the service profile) is already undertaken by initiatives like CC/PP.

However, there are some important issues to notice in this model. Since all interfaces between storage of personalized information and service logic are standardized, and since it is not defined what the SharedLocation consists of, this model can support personalization of both service data and service content.

To enable personalization of service data, the SharedLocation can be accessed through any Inter Process Communication (IPC) mechanism, for example through shared memory. To enable personalization of service content, the SharedLocation can be a hard disk accessed through disk I/O operations.

B. Model 2

Fig. 5 displays a model for personalization of services without using standardized interfaces. It works by adding a personalization layer in the same location as the service logic of

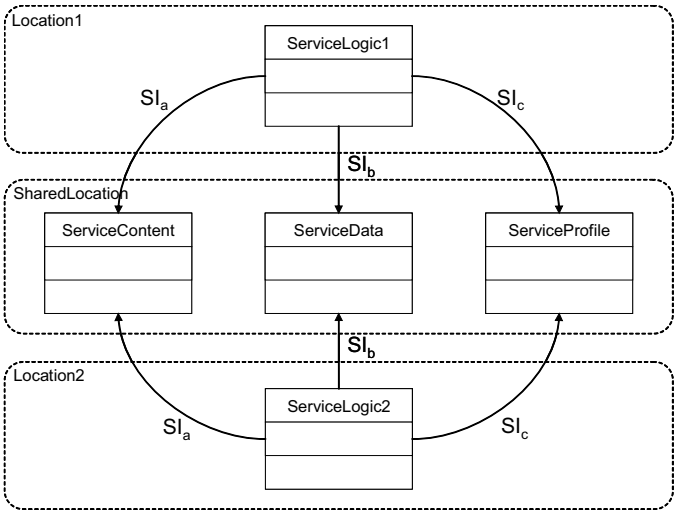

Fig. 4. Standardized interfaces for accessing personalization information

the original service. This layer is used by service logics in other locations to access the personalised information in the original location. This model is *asymmetrical*, since it is only the original service in the original location that access personalized information in the usual manner. Other service logics in other locations only access the intermediary layer, which implements a subset of the original interfaces.

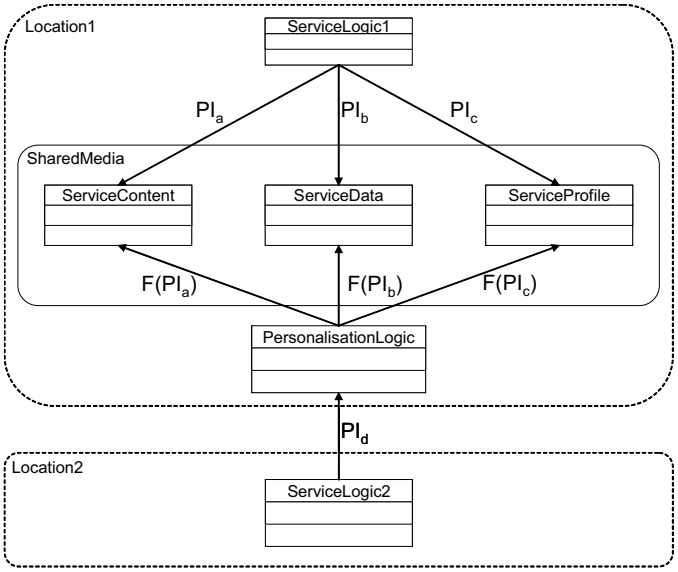

Fig. 5. Personalization without standardised interfaces

The approach for personalizing services without using standardized interfaces works by *interception* of relevant

information. An additional 3rd party logic component is installed in the system where a service logic resides (PersonalizationLogic). This component implements some interfaces $F(PI_{a-c})$ which are based on (a function of) the interfaces PI_{a-c}, which are proprietary. The reason these interfaces are functions of the proprietary ones are that they most likely only provide a subset of the functionality provided by the original interfaces; this is because they are based on *observations* (of behavior) and not *specifications*.

However, as a user moves from Location1 to Location2 and reinitiates service access through ServiceLogic2, this component can communicate with PersonalizationLogic to retrieve personalization information from the previous location through another proprietary interface called PI_d. The SharedMedia component can for example be the file system of a personal computer or a registry (as in Windows); information that allows personalization of services is typically persistent and thus stored in a persistent database or similar.

Opportunities and Challenges with Model 2

By applying this model, it can be possible to personalize already existing services, as opposed to model 1, which only allows future services to be personalized. This is therefore an opportunity to improve already existing services and ensure continued and maybe increased use of such.
However, the major drawback with this model is that not all features of all services can be personalized because the existing interfaces may be too closed. Implementing personalization with this architecture requires some reverse-engineering, as long as interfaces are not open and published (typically the case with applications in MS Windows).

More importantly, personalization of information that resides in the Service Data component will in most cases be impossible; this memory range is outside reach for other components to access. This heavily dictates the type of personalization enabled by this model and decreases the feasibility for achieving service continuity.

C. Summary and Comments on the Two Models

Model 1 can support extensive personalization by enabling sharing of personalized information in both the Service Data and the Service Content component. Model 2 only directly supports personalization of a subset of the information carried by the Service Content component. Thus, depending on the personalization needs of the service, and what type of service it is, the most adequate model should be selected. However, to enable and improve personalization of future mobile services, the first model should be selected and further work should be carried out towards defining standards that allow sharing of personalization information between different instances and implementations of the same or similar services.

VI. Case studies

A simple yet illustrative case study will now be discussed in light of the proposed personalization models/approaches. In both scenarios, it is assumed that a user accesses a service in one location, then moves to another location and resumes operation of the same *type of service* using another *implementation of the service*. For a typical Internet browser like Mozilla Firefox, the following elements could be subject to personalization:

- bookmarks list
- history list
- saved passwords
- current URL
- bookmarks toolbar (quick access to web pages)
- default download folder
- saved form information
- cookies (e.g. to preserve identity)
- popup windows blocking activated?
- Etc.

The listed features are related to the *behavior* but not the *look and feel* of the service.

A. Personalizing an Internet browser using Model 1

First, the entities that must be considered to allow personalization of this service must be identified. An Internet browser typically allows a user to browse pages on the WWW. To accommodate personal requirements, a bookmark list and quick access bars are commonly employed. Also, as long as a browsing session has been initiated, there is always a current URL in the address field. Also, browsers commonly keep history of visited pages. Some of the service content is accessed through a standardized interface already, the HTML code.
Personalizing this service means allowing the user to access bookmarks created and stored by this service from other locations by potentially other implementations of an Internet browser. Moreover, it means that also the other elements listed above should be accessible from other locations. By using Model 1, this means two things. First, there must exist a shared media where bookmarks, quick access bars and other information can be put by the first service implementation at service closedown and be retrieved by the second one at service startup. This could for example be a file server (e.g. ftp server) provided by a third party. Second, the interfaces towards this shared media must be known by both implementations, i.e., the protocol and messages for accessing the elements as well as the format they are stored in must be known and agreed upon by both implementations.

B. *Personalizing an Internet browser using Model 2*

Several applications already exist to allow users to synchronize bookmarks towards a remote server, in such a way that they can be almost ubiquitously available independent of the current network and device. This is primarily enabled using a standardized format for storing bookmarks, namely the XML Bookmark Exchange Language (XBEL) [12]. Extensions for the Mozilla Firefox browser are readily available[1].

However, bookmarks are not the only information that can be used to personalize an Internet browser. Is it possible to intercept the current URL, so a session can be transferred from one running browser to another? Or when using a "tabbed" browser, can all the tabs be available in the new location? The current URL is, as defined by the composition earlier, part of the service data. Using Mozilla Firefox as an example, the bookmark and history list can be found in the file system. However, parsing these files, although text based, requires knowledge of the construction and format.

The bookmark list is in fact an HTML document that can be edited. It is possible to transfer this list to another location, potentially transforming it into another format and then reinitiating a browsing session in another location with the same bookmark list.

Applying Model 2 to personalize an Internet browser means installing a third party service logic component in the location of the original service, where this new component can access the information that is needed to personalize the service through the filesystem or another persistent storage component (like the registry in MS Windows). On a standard Windows platform, this component could read the bookmarks from the filesystem. The bookmarks must then be made available to other instances and implementations of an Internet browser. This could be performed by either allowing direct connections to be made from other service logics towards this PersonalizationLogic component, or the PersonalizationLogic could be responsible for uploading this information, in a format recognized by the service in the second location, to a location which is known to be accessible by the second service instance/implementation. The former approach might prove difficult, since access towards a computer in a home LAN often is restricted by firewalls, thus disallowing access to this information from a remote location.

VII. CONCLUSION

This paper contributes to the elucidation of the personalization of services and its meaning to the user. It presents an overview of the information space that must be considered and covered to realize service personalization. The step-by-step general process of personalization is also given.. Most importantly, two models to implement personalization are presented and their opportunities and challenges are also discussed. Whereas one of the models can support advanced personalization, the other one is restricted to enable personalization of only a subset of the functionality provided by a service.

However, in order to realize personalization, some other issues must be considered. The first issue is the format of Service Data, Service Content and Service Profile that needs to be flexible and expandable. There are yet no standard for representing these, because there is no general agreement on what these profiles should contain. The second challenge with Service Data, Content and Profile is the storage location. They must be stored in a location where it is accessible from any service logic component that should be personalized. This leads to the third challenge, which is the ownership of the profiles. If users have problems trusting service providers, users should be the owners of the profiles and they should be stored in a way that ensures and protects their privacy.

[1] http://addons.update.mozilla.org

REFERENCES

[1] Blom, J., "Personalization – A Taxonomy", Conference on Human Factors in Computing Systems (CHI). Hague, Netherlands, 1-6 April 2000. ISBN:1-58113-248-4

[2] Jørstad, I., van Do, T., Dustdar, S., "Personalisation of Future Mobile Services", 9th International Conference on Intelligence in Service Delivery Networks. Bordeaux, France, 18-21 October 2004.

[3] 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, Service Aspects, "The Virtual Home Environment (Release 5)", 2002.

[4] Roque, R., Soares, T., and Oliveira, J., "VESPER Project – Validation of VHE Concept. 2001", Available online at: http://citeseer.ist.psu.edu/459241.html

[5] VESPER Project, IST-1999-10825, "D41 – Demonstration Services Requirements", January 2001

[6] Houssos, N., Koutsopoulou, M., Schaller, S., "A VHE architecture for advanced value-added service provision in 3rd generation mobile communication networks", Service Portability and Virtual Customer Environments. IEEE, 2000.

[7] Lankhorst, M.M., Kranenburg, van H., Salden, A., Peddemors, A.J.H., "Enabling Technology for Personalizing Mobile Services", Proceedings of the 35th Hawaii International Conference on System Sciences, 2002

[8] Kramer, J., et. al., "A User-Centered Design Approach to Personalization", Communications of the ACM, Vol. 43, No. 8, August 2000

[9] Nielsen, J., "Alertbox for October 4", 1998, http://www.useit.com/alertbox/981004.html

[10] Jørstad, I., van Do, T., Dustdar, S., "An analysis of service continuity in mobile services", 2nd International Workshop on Distributed and Mobile collaboration (DMC), WETICE conference, 14-16 June, Modena, Italy, 2004, IEEE Computer Society Press. ISBN: 0-7695-2183-5

[11] Jørstad, I., Dustdar, S., van Do, T., "Towards Service Continuity for Generic Mobile Services", The 2004 IFIP International Conference on Intelligence in Communication Systems (INTELLCOMM 04), Bangkok, Thailand, 23-26 November 2004. ISBN: 3-540-23893-X

[12] Drake, F., "The XML Bookmark Exchange Language, Release 1.0", October 28th 1998, available online at: http://pyxml.sourceforge.net/topics/xbel/