Applying Profile Views to Bridge Between Different Personalisation Architectures

Bjoern Wuest, Olaf Drögehorn, Klaus David

University of Kassel, Chair of Communication Technology
Wilhelmshöher Allee 73, 34121 Kassel, Germany
{bjoern.wuest, droegehorn, david}@uni-kassel.de

Abstract: User profiles are typically stored locally within proprietary personalisation architectures at service providers. Replicating the same information of the user profile, e.g. email address, across multiple independent service providers decreases consistency of user profiles. Centralising or exchange profile information increases consistency of profiles. Law issues, company policies and proprietary profile structures prevent successful profile exchange between service providers. Existing solutions for centralised profile storage like Microsoft Passport or Liberty Alliance are limited to specific information in the user's profile, e.g. authentication information. This paper presents a profile architecture to keep the user's profile in his domain. Service providers access the user profile by defined translations between the service providers profile structure and the user's profile structure. The translations defined are adaptable to reflect modifications in the different profile structures, e.g. Amazon, eBay etc.

1 Introduction

Personalisation has the potential to increase the value of web services for the user. Examples of personalised web services are Amazon, eBay, New York Times etc. Profiles are a cornerstone of personalisation. Profiles keep the user's preferences and gathered habits. Web services use the information in profiles to adapt to the user [HBD00][CDA00].

Personalisation architectures, where profiles are a part of, are mostly proprietary systems. Interoperability and exchange of profile information between personalisation architectures is not foreseen. Instead, an individual profile database is kept with each service provider. This raises consistency issues of profile information by replicating the same information throughout a number of service providers.

A simple scenario may illustrate the problem of consistency. Assume two service providers offering personalised web services, one online book store and one online auction system. Both services provide a newsletter service with latest information being sent to registered users via email. Further, both web services support adaptation of font style and font size for different text styles and require the user's credit card information for offering paid services. Both service providers act independently and have no interest to share their data. Privacy policies and laws may prohibit data sharing. A user registered at both services must setup his email address and preferred font settings with both web services at both service providers. If any information of the user is changing he must update his information with both service providers. In a real world scenario the user may

not only register with two systems but with dozens, possibly hundreds of services, increasing the effort to keep his information consistent among all of these services.
The Microsoft Passport service [MNP] and the Liberty Alliance [LAP] are efforts to keep a service independent user profile. However, both are limited to user authentication and related information such that font settings and the like are not covered by these two efforts.
Synchronisation mechanisms such as SyncML [OMA] or xmiddle [Mas02] reconcile distributed profile information. However, synchronisation of profile information is impractical when used for synchronising and distributing profile information between different independent service providers. Reasons for this impracticality are local privacy policies and laws, competition and incompatible profile structures due to missing standards. We understand that synchronisation mechanisms are still necessary, but only within the user's domain, where those reasons are of no concern. We do not agree that synchronisation is applied between two parties without the owner of the data being involved, in our case the user of the service providers' services; although our approach does not enforce avoidance of such synchronisation.
In this paper we introduce an approach to address consistency of information for the scenario above. The goal is achieved by storing profile data within the user's domain. Service providers link to the information in the profile. Keeping profile information within the user's domain is proposed in [RBG01] but their approach lacks of the flexible profile structure expected by the different service providers. Improving consistency of profile information all together with flexibility in profile structure are both being addressed within this paper.
In section 2 we give a clear and strict definition about how a profile is organised. This definition is tailored towards separation of profile information and profile structure. Its possibilities and advantages are discussed in section 3. Section 4 provides guidance on how to utilise the separation introduced in real world solutions. Finally the conclusion is given.

2 Definition of profiles

A profile is a collection of information. In a user profile this information reflects the preferences and gathered habit of the user. The information is accessed by a well known key. To represent a key a keyword or description or any other form able to uniquely identify information may be used. A pair of key and information is called a profile entry. Every profile entry may occur only once in a profile, meaning that no key may occur twice in a profile.
A profile consists of several profile entries. A telephone book is such a profile. The contacts in the telephone book profile are pairs of names and telephone numbers. These pairs form the profile entries with one profile entry per contact. The name of a profile entry is the key to the profile entry while the telephone number is the information of the profile entry.
Profiles may contain Meta information to augment information. Such Meta information could be used to constrain the information, e.g. to enforce a proper format of phone numbers and dates. Other Meta information could limit access to profile information. Meta information may also semantically describe profile information. Augmenting the

profile with attributes provide support for Meta information in a profile. Every profile entry may be augmented with any number of attributes. Attributes may be shared among different profile entries of the same profile and of different profiles.
In the telephone book 'home' or 'business' or 'mobile' are attributes. The simplicity of attribute augmentation proposed in this paper is sufficient for the purposes presented later in the paper. More advanced attribute augmentation schemes, e.g. basing on ontology, may be defined without affecting the results of this paper.

3 Separation of profile structure and profile information

To increase data consistency of profile information we regard it to be useful to keep the profile information in one profile, within the domain of the user, under his control. Yet, this raises the problem of how proprietary personalisation systems of service providers, which are used to keep their own profile, retrieve and use information in the user's profile. The identified challenges of storing the user's profile within the user's domain are unavailability of the user's profile when *disconnected* from the user's domain on the one hand and the different standards of profile structure when the user's profile is stored in only one profile structure on the other hand. We address the problem of *disconnection* from the user in section 4 and concentrate on how to solve the gap arising with different profile structures in this section. In this section we make the assumption that service providers have access to the profile information stored in the user's domain every time they require access. In the following section we drop this assumption but for now it simplifies the considerations following.
In the scenario of the online book store and the auction system, both personalisation systems use their own proprietary profile structure. The different profile structures would require for different, independent profiles. This requirement becomes obsolete once the structure of a profile is separated from the information stored in the profile. This separation allows defining several different structures on the same information stored in a profile. Instead of duplicating information stored in the profile the information is stored only once and within the user's domain. The profile structure is kept with the service providers. A linkage between the profile structure located with the service provider and the profile information located with the user is established. This linkage uses a plain, simple yet sufficient mapping between structure and information, e.g. unique identifiers. This linkage is transparent to both the user and the service provider. Further the linkage is free of context and semantics. Context and semantics of profile information is addressed by the profile structure used by the service provider. The meaning of profile information is brought in by the profile information by itself.
Thus, there are profiles and views defined on profiles. Profile entries no longer contain a key but instead a unique identifier. Views contain keys mapped to unique identifiers. The unique identifiers in a view map to a single profile entry.
Views are limited to one profile. Further research may investigate if it is useful and suitable to extend views to be backed by multiple profiles. For the targeted scenario where the user's profile information of several service providers is pooled in one profile and different profile structures are defined on this one profile this limitation has no negative effect.

The remaining of the paper discusses how the proposed profile architecture with separating profile information from the profile structure may be used in the real world. The increase in privacy with reduced requirements towards trust and gained consistency of profile information is demonstrated. Further, a user controlled yet automatic algorithm for translating between different profile structures, the views, is given.

4 Application in the real world

In the previous section we postulated that a separation of profile structure and profile information is beneficial towards consistency. In this section we demonstrate the advantages of the separation presented above by applying it to the scenario of proprietary and isolated personalisation architectures of the introduction.

In our scenario there are two service providers offering web services, one online book store, and one online auction system. Both service providers keep behavioural and configured preferences of the user in profiles and personalise their service towards the user's preferences. Both service providers use their own proprietary personalisation architecture which is optimised to their offered service. Yet, both service providers depend on identical information about the user like his email address, postal address and credit card information. An exchange of information between both service providers is not possible, not wanted and not allowed due to technical incompatibilities, company and privacy policies and laws.

Traditionally the service providers keep the information about the user in their domain. We move the information from the service provider's location into the user's domain, e.g. his terminal he uses to access the web services offered. The service providers keep the structural information about profile information within their domain. Data consistency is improved since profile information about the user is kept only in one place, in the user's domain, and not spread and duplicated among all service providers the user is registered to. To keep profile information consistent within the user's domain we foresee the use of synchronisation middleware, e.g. xmiddle [Mas02].

Questions arise on the case where service providers have interest to keep information about the user, e.g. his email address, to send email notifications or advertisements. A common solution to this problem is caching of such interested information. For updating the cache information, simple concepts such as synchronisation the cache every time the user connects to the service, may be applied. This automatic update of cache information might also be a faster and more reliable solution than to ask the user to manually navigate to every single service he is registered with and update his information.

As seen in this section the separation of profile information and profile structure has huge impact on how profile data may be managed and maintained. Consistency of profile information is increased even caching is used by avoiding duplication of identical profile information among all service providers the user is registered with. Further, automatic profile reconciliation increases the convenience of use of personalised systems by leveraging the user from updating his profile information manually in many places. Instead the user updates his information in only one place, e.g. in his domain.

5 Conclusion

Personalised services increase their value by adapting themselves to the user. This adaptation is based on profile information about the user, kept at the service provider. Exchange of user profiles between service providers is not done because of legislative provision, company policies and proprietary profile structures. Centralising of profile information, as done e.g. by Microsoft Passport and Liberty Alliance, addresses only specific parts of the user profile, e.g. user credentials. In the paper we proposed to keep user profiles in the user domain and to separate the profile data from the profile structure.
Keeping user profiles in the user's domain and letting service providers access the user profile overcomes law and company policy restrictions. Separating the profile data from the profile structure allows defining different profile structures, used by different independent service providers, on the same profile data. This separation of profile data from the profile structure benefits consistency of data by avoiding redundancy and improves interoperability between different proprietary personalisation architectures by providing a flexible translation technique based on attribute dictionaries. A definition of profiles is used to show avoidance of redundancy of profile information. Profile representation translation is derived to introduce the presented research results in real world systems.
The viability of separating profile information from profile structure is shown by an example where user profile information (email address) is shared across different independent web service providers. With this separation, users need to update their information only once. There is no more need to update information with all web service providers the user is registered at.

This work was supported in part by the German Ministry of Research bmb+f Wireless Internet Zellular and MIK projects. The paper represents the work and contribution of individual parties involved in the project. We thank our project partners for the good and fruitful discussions.

Bibliography

[CDA00] Cingil, L., Dogac, A., Azgin, A., "A broader approach to personalization", In Communications of the ACM, Vol.43.8, p. 136-141, August 2000
[HBD00] Hirsh, H., Basu, C., Davison, B. D., "Learning to personalize", In Communications of the ACM, Vol.43.8, p. 102-106, August 2000
[LAP] Liberty Alliance Project, http://www.projectliberty.org/
[Mas02] Mascolo, C., Capra, L., Zachariadis, S., Emmerich, W., "xmiddle: A Data-Sharing Middleware for Mobile Computing", In Proceedings of the 24th International Conference on Software Engineering, Orlando, Florida, USA, 2002
[MNP] Microsoft .net Passport, http://www.passport.net/
[OMA] The Open Mobile Alliance, SyncML, http://www.openmobilealliance.org/syncml/
[RBG01] Riche, S., Brebner, G., Gittler, M., "Client-Side Profile Storage: a means to put the user in control", Public Technical Report, Hewlett Packard Laboratories Grenoble, November 2001

Towards Update Relevance Checks in a Context Aware Mobile Information System

Hagen Höpfner
hoepfner@acm.org
International University in Germany
School of Information Technology
Campus 3, 76646 Bruchsal, Germany

Abstract: In order to reduce transmission cost mobile information system clients often cache data retrieved in a context aware manner. In the case of updates it may happen that this data becomes outdated. So, the caches must be invalidated. In this paper we discuss how to check the relevancy of updates regarding the server database as well as the client contexts.

1 Introduction and Motivation

Mobile information systems are often client/server systems with mobile clients that request information via a wireless connection. In order to reduce the communication costs, mobile devices cache received data. If data on the server changes, one has to inform the mobile clients that hold or should hold modified data. In [Hö05, HSS04] we presented approaches to check the relevancy of updates on the server side. We considered conjunctive queries with inequalities and focused on the relational data model. But mobile information systems have to consider the context of their usage. In [HS03a] we discussed a general context model that allows to hoard data automatically on the mobile client without formulating an explicit query. In this paper we present first ideas of combining both approaches having regard to the update relevancy check. The long-term objective is a context aware mobile information system, that handles conjunctive database queries with inequalities. The vision behind our work was presented in [HS03b].

The remainder of the paper is structured as follows. Section 2 is a summary of the foundations of this paper. In Section 3 we discuss the server side relevancy check for context aware conjunctive queries with inequalities. Finally the paper closes with a summary, conclusions and an outlook on ongoing research in Section 4.

2 Foundations

In this section we recapitulate some ideas of the used context model but restrict to aspects necessary for the understanding.