

Mobile personalisation: new challenges for privacy

Greg Darke, Judy Kay, and Bob Kummerfeld

University of Sydney, Australia

Abstract. One of the challenges of mobile and pervasive computing is to enable ready access to personal information, particularly user models, which can drive the local personalisation of information on mobile devices. At the same time, people may want to ensure that sensitive parts of their user model and other personal information is restricted to selected devices that the user trusts with that particular information. This paper describes the motivation for new approaches to privacy management for mobile and pervasive contexts. We describe a design that takes account of dual demands posed at a technical level and the need for a foundation for creating user interfaces to control privacy of user models.

1 Introduction

There is a tension between the two competing goals of ensuring privacy on mobile and pervasive devices at the same time as providing the ready availability of user information, particularly user models. Privacy goal call for a mechanism that ensures the user can restrict sensitive parts of their information to devices that they trust with that level of particular information.

It is clear that user interfaces will be challenging to build for several reasons. One of these is related to the complex nature of the actual privacy needs of users: these mean that it is important to be able to create quite flexible and changeable privacy policies or rules. This is well described in the seminal paper on the dynamic nature of privacy ¹, where the authors describe the tension between keeping information hidden and, at the same time, ensuring it is usefully available. They carefully map out the ways in which privacy preferences are very fluid, for example, needing to be updated in response to changes in the context of the user. As they note, this is in stark contrast the common misconception that privacy is just the strict enforcement of a set of static rules.

We illustrate these ideas in a scenario that we will refer to throughout the paper.

Albert is an IT professional. He is interested in pervasive computing, and keeps up with the latest news by reading a collection of RSS¹ feeds. He also subscribes to other feeds, one from his work on a new secret product he is working on, the other containing information regarding the home

¹ Really Simple Syndication

automation project he is working on at home. Albert reads news feeds on his smart-phone on the commute from work to home. Later that night, at his home desktop, he reads more news, but clearly does not want to see articles that he has already read. When Albert returns to work the next day, he reads news there, but (to avoid IP problems) does not want to see the feeds about his home automation projects.

There are several existing products that provide this service, including Google Reader², NewsGator³, and many others. However, all products make use of a central system. In our scenario, Albert may not feel comfortable with all this information about him being held in a central system that is outside his control and he is unsure about trusting the organisations with the user model. This means that for a user like Albert, such *cloud computing* solutions are inadequate.

There are many emerging cloud computing possibilities, providing services that previously existed only as local applications on a user's computer. Some of these services may contain personal data, most likely including user models, and these may include potentially sensitive and highly personal information such as that extracted from calendars or medical data.⁴ While people may value the services offered by such cloud services, many may be concerned about the privacy aspects. For example, they may not wish their data to be mined to provide targeted advertising. Worse still, there is the risk that it is leaked.

We want access to personalisation and our user model on our desktops and carried devices without a relying on a third party as envisaged by cloud computing. In the next section, we review the properties of the two main architectural approaches to storage of user models. Then, we discuss the challenges for supporting privacy, followed by an outline of our approach.

2 Architectural approaches

The architectures for storing and managing user models can be along a continuum, with pure centralised systems at one extreme and highly distributed architectures at the other. Table 1 summarises desirable properties for such a system, showing the trade-offs associated with each of these extremes.

The top three rows of the table deal with the service aspects. The central user model, common to many web services, has many advantages. From the user perspective, the most notable of these is that it can provide a consistent view of the services, no matter how the user connects to the service. Centralised systems are likely to be easier for the programmer to implement and the cost of ensuring high levels of reliability can be shared across many users. However, as shown in the third row, centralised solutions do not deal well with disconnected operation, for example, when the user is out of range of wifi.

² <http://reader.google.com>

³ <http://www.newsgator.com>

⁴ eg. www.google.com/health

Table 1. Desirable properties of User Models

Desirable	Central	Distributed
Consistent view of service	+	-
Ease of implementation and support for reliability of service	+	-
Able to work when disconnected	-	+?
Access to user model	-	+
Control of release	-	+
Control over location of model	-	+
Control over aggregation	-	+

The remaining four rows deal with the user aspects. The problems with a centralised architecture follow from the fact that the user model is not controlled by the user: they do not have direct control over who the information is released to, and how this information may be aggregated.

The pure distributed model has almost the opposite strengths and weaknesses. Each client has a independent user model, which can potentially give the user more fine-grained control over the release of information: each part of the user model can be released separately. Similarly, there is potential for the user to control the aggregation of their data. However, this introduces potential problems in maintaining a consistent view of the service, as in the scenario. The case of disconnected operation is less clear. On the one hand, the user may be able to achieve their tasks on a disconnected machine. For example, if Albert in the scenario is on a train, with no network connectivity, but his phone has pre-loaded news and the relevant user model, he can work with these. Of course, if this model is not up to date, he may be frustrated by being presented with news he has already read.

3 Challenges

One of the core challenges for ensuring that users can control the privacy of their user models comes from the difficulty of creating effective user interfaces. This proved to be a problem for P3P which aimed to define a policy which, once set up, could operate without bothering the user. At the other extreme, Langheinrich proposed that a pervasive system should issue notifications 2, 3, whenever there was the potential for a privacy violation.

MovieLens 4 Caching? 5

4 An Accretion/Resolution Approach

better to explain the broad approach in terms of an example as we discussed and better to explain a few cases in more detail and leave others

- We propose to use the AR (*Describe AR here*) method of modelling the UM (as described above (*back reference to description*)).

- We will use a distributed UM (one on each device)
- Evidence will be delivered to the UMs on other devices (the synchronisation step)
 - The propagation of information from one UM to another depends on the privacy rule of the context (*is this the right word? - yes it will be fine*).
 - These rules state which (device, application) pair IS allowed to get this information. *** NEED AN EXAMPLE
 - Default rule is that evidence may only stay on the (device, application) pair in which it was created
 - User can then select if they want to propagate the evidence to a particular pair, or to a *group of IDs*.
 - The user can select to allow information to go to all devices (to allow sharing of information with users nearby).
 - These data propagation rules can be added directly to a piece of evidence (to allowing sharing of either specific news items with others, or to hide the information from specific devices)
- Due to the way that AR works, once the information has been propagated to the respective models, there will be a 'Consistent view of service' (Which is one of our desirable services).

References

- Palen, L., Dourish, P.: Unpacking “privacy” for a networked world. In: CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2003) 129–136
- Langheinrich, M.: Privacy by design – principles of privacy-aware ubiquitous systems. In: Ubicomp 2001: Ubiquitous Computing. Volume 2201 of Lecture Notes in Computer Science., Springer (2001) 273–291
- Langheinrich, M.: A privacy awareness system for ubiquitous computing environments. In: UbiComp 2002: Ubiquitous Computing. Volume 2498 of Lecture Notes in Computer Science., Springer (2002) 315–320
- Miller, B.N., Albert, I., Lam, S.K., Konstan, J.A., Riedl, J.: Movielens unplugged: Experiences with a recommender system on four mobile devices. In: IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces, New York, NY, USA, ACM (2003) 263–266
- Helal, S., Hammer, J., Zhang, J., Khushraj, A.: A three-tier architecture for ubiquitous data access. In: Computer Systems and Applications, ACS/IEEE International Conference on. 2001. (2001) 177–180