Proceedings of the Third IASTED International Conference
ADVANCES IN COMPUTER SCIENCE AND TECHNOLOGY
April 2–4, 2007, Phuket, Thailand
ISBN CD: 978-0-88986-656-0

CONFLICT BETWEEN PRIVACY AND PERSONALISATION IN A PERVASIVE SERVICE ENVIRONMENT

M. Howard Williams, Elizabeth Papadopoulou, Nick Taylor, Sarah McBurney
Heriot-Watt University
Riccarton, Edinburgh
UK
{mhw, ceeep1, nkt, ceesmm}@macs.hw.ac.uk

Kajetan Dolinar
SETCEE
Ljubljana
Slovenia
kajetan@e5.ijs.si

ABSTRACT

For a pervasive system to function effectively in a world in which the user is surrounded by a large number of heterogeneous computing devices and communication systems, it is essential to provide adequate support for the user. For this personalisation is the key element, both for improving the user experience and for enabling it to function without the need for unnecessary interaction with the user. In pervasive systems personalisation can take a number of different forms, and four different forms of personalisation that have been incorporated into the pervasive system developed in the Daidalos project are described. The paper also describes two different approaches used to protect the privacy of the user in the two phases of Daidalos. However, there is an interaction between personalisation and privacy and some of the issues involved are discussed. These also affect proactive behaviour of the system. They are currently being used to plan the development of the second phase of the Daidalos pervasive system.

KEY WORDS

Privacy, personalisation, pervasive services and virtual identity

1. Introduction

The development of pervasive computing environments arises in response to the growing complexity facing the user from the rapid expansion in different communication networks and devices and the growing diversity of services that are available. The aim of pervasive computing is to support the user and enable him/her to take advantage of these developments by providing ubiquitous access to services while protecting the user from their underlying complexity [1]. Such systems need to be context aware and able to adapt their functionality and behaviour in response to changes in their environment [2]. The need to handle the growing variety of heterogeneous devices and services coupled with their personalized and integrated use via appropriate networks has motivated a number of research efforts (e.g., [3, 4]).

The Daidalos project [5] is a large European research project (involving some 45 partners) whose aims include the development of a pervasive computing environment that provides users with dynamically adaptive services to support both stationary and mobile users. Both personalization and context-awareness have an important role to play in this in order to provide the user with a high-quality service that will “best” serve their needs [6].

However, another goal of the Daidalos system is to provide adequate protection of user privacy. This is not always easy and the techniques used to support personalization can easily conflict with the requirements for privacy. This paper describes the kinds of personalization being implemented in the Daidalos platform and some of the issues relating to privacy. It also describes briefly similar problems encountered with proactive behaviour. These are currently being used in the redesign of the Daidalos platform.

2. Personalization and Pervasive Environments

Personalization is generally regarded as the collection of processes that are used to adapt the behaviour of a system so that it appears differently to different users or even to the same user under different circumstances in accordance with their individual user preferences.

The focus of most research in the area of personalisation has been on information retrieval on the Web. One of the aims in this case has been to use information about the user to select information that is likely to be of most interest to the user. The user preferences and current user context are generally used to control this process. Some examples are given in [7]. Another aspect of this research has been to adapt user queries to retrieve information that is most likely to be relevant to the user. For example, by monitoring choices made by the user in searching the

Web, one can use the resulting knowledge to disambiguate queries and speed up future searches.

Another facet of personalization is its use in the control of layout and presentation for applications. Yahoo provides the classical example of this and other software applications have followed this lead. Users can specify how they want the screen to be laid out, as well as preferences for background, size and colour of fonts, etc.

Similar types of personalisation can be found in pervasive service environments. One example is the Tivoli Personalized Service Manager [8] developed by IBM, which provides an integrated infrastructure of software products for Internet service provisioning. It enables the generation of web pages for specific devices, personalisation of portal home pages, and provides services such as calendar, agenda and address book which can be used to develop additional services. It also supports translation into different languages. However, it is somewhat limited in that it only takes account of the user's profile and preferences and not dynamic context information such as location or current activity, and its personalisation features are restricted to the simple services supplied by the product itself.

The SPE (Secure Persona Exchange) framework described by Brar and Kay [9] provides personalized services to users in ubiquitous computing environments based on user preferences stored on mobile devices although like [8], it does not take account of dynamic contextual data while achieving personalization.

Another area in which personalisation is used in pervasive service environments is that of service composition. For example, Sheng et al. [10] describe a personalised composite service specification architecture, which enables users to specify their needs through a set of process templates. However, this approach demands a heavy overhead of the user when orchestrating a composite service, which is not realistic when a large number of service compositions may occur dynamically.

'eFlow' (Casati [11]) is a system that incorporates a form of dynamic service composition, which includes personalisation based on user input of their requirements. The composed services are not technically dependent on each other, although they do complement one another. The dynamic composition provided may be changed over a period of time, and hence over many compositions rather than within a single composition. Real dynamic re-composition based on continually changing user requirements is not addressed.

Other projects researching personalisation in ubiquitous environments include Carnegie Melon's Project Aura [12], MIT's Project Oxygen (Intelligent Room), Portolano, Sentient and others.

Project Aura addresses self-adaptation based on user intent, thereby taking account of the task that a user wishes to perform [13]. Task requirements consist of the services needed to perform the task and the associated user preferences. Using these, the system decides how to configure or reconfigure the intelligent environment to best support the user in this task.

MIT's Project Oxygen focuses on the Intelligent Room, which is a type of Intelligent Environment (IE) whose ultimate goal is to enable computers to communicate with users through vision and speech as humans do [14]. To achieve this goal, the Intelligent Room relies on activity based context modelling which involves creating a high-level representation of the context of the current users involved in that activity.

Thus it can be seen that there are many ways in which personalization can be used to adapt the behaviour of a system, and in a pervasive environment there is scope for a variety of different forms of personalization. In Daidalos we have focused on four of these.

(1) Personalization in Service Selection. When a user requests a service, the Pervasive System Platform uses a classical Service Discovery approach to find possible services that can be used to satisfy the user's request. If more than one possible service is found, the list of options is ranked using the user's preferences and current context and the most appropriate one is selected. This aspect of personalization is referred to as Personalized Service Selection. This is described in more detail in [15].

(2) Personalization of Component Services. If an individual service component allows itself to be personalized by the user, then the particular user's preferences (taking account of his/her current context) can be passed to the component via a set of parameters in the script. This form of personalization is application specific and the Pervasive Service Platform merely provides the parameters required without any understanding of the component.

(3) Personalized Call Redirection. In the case of communication services such as phone calls and messaging services, one has a particular problem relating to how and when the connection is made. In a pervasive system where the aim is to give the user more control over his/her environment, it is not unreasonable to expect the called user to have some control over this process, specifying when, where and from whom calls or messages can be received. For example, if one is at home, there are times when one does not want to be disturbed by a work phone call at all and others when one would accept such a call if it is from one's boss but not otherwise. More details of this are given in [16].

(4) Personalized Network or Device Selection. In assembling a number of component services together to

meet a user request, this may include a network service and/or a particular device service (such as a print service). Having selected a particular network service, as the user moves about or other users load the network, the characteristics of the network may change and the Quality of Service (QoS) available to the user may fall. If this happens the pervasive environment needs to check whether other options are available and, if necessary, switch to an appropriate one. Once again user preferences and context need to be taken into account in this process.

In each of these four cases one needs to maintain for each user a set of context-dependent user preferences. In general these preferences may be distributed or replicated across different machines provided that the appropriate preferences are accessible wherever they are needed.

3. Privacy

In a pervasive system environment such as this there are three main areas where the privacy of the user needs to be protected, namely:
(1) Protection of information on the services that are run by the user. When a user requests a service, the Pervasive Service Platform assembles the components necessary to meet that request. It is important therefore that no other service supplier is aware of what services the user is using. Moreover, even those service suppliers who are providing service components currently being used by the user should be given as little information about the user as possible. Clearly some form of user identification is needed for billing purposes but beyond this the service supplier should not be able to identify the user. The user may even wish that separate invocations of the same service in different contexts may not be connected by the service supplier. Thus if the user uses a service from home, he/she may not wish it to be linked with any use of the same service at work.

(2) Protection of context information. The most obvious attribute of context information that is important in a pervasive system is that of location. The location of the user is an essential but sensitive piece of information that, in general, the user would not wish to be released to anyone unless permitted by the user. Other aspects of context are also sensitive. For example, another attribute of interest is the user's current task or activity, which may be inferred from other attributes. Equally sensitive may be who the user is with at any point in time. For the sake of the privacy of the user access to these attributes needs to be securely protected.

(3) Protection of personal preferences. The user preferences are a set of context-dependent rules that are used to tailor the behaviour of the system to the user's needs and are as sensitive as context information. In particular, knowledge of the user's preferences could be used to infer context information such as location. For example, if one knows the call redirection rules for another user and one's call is not answered, one may be able to infer the location of the user. Likewise, such knowledge could be used to build up a picture of the user's service usage.

In order to handle the first problem, the protection of information on services, Daidalos is investigating the use of virtual identities. Each user may have a set of virtual identities (VIDs), which cannot be linked by third party service suppliers. In this way the anonymity of the user can be preserved as far as necessary and, by using different VIDs when accessing the same services, the service supplier is unable to associate the different uses made by the same user.

Once this is achieved, by placing appropriate security controls on the access of context and user profile information, the other two aspects of privacy can be assured too.

4. Handling privacy with role-based VIDs

In the first phase of the Daidalos project VIDs were handled using a role-based approach. The user could set up a small set of VIDs, each allocated to a specific user role. For example, a user might allocate one VID for use when working, a second for use at leisure (at home or out and about) and a third when on holiday. Obviously this could be extended to cover whatever appropriate sets the user required.

Associated with each VID is a set of user preferences. These have to be set up manually by the user. Ideally one would like to simplify the task for the user and incorporate some mechanism to assist the user in updating user preferences by having some information on the set of VIDs belonging to the user. This would enable the system to transfer sets of user preferences between VIDs, or when a new preference is identified the system could ask the user whether it should be added to all sets of preferences, some subset of them or only the current one. However, in order to maintain strict security no information on the linkage between VIDs may be kept in the system. This means that the user must maintain each set of preferences independently. For this reason it was envisaged that the number of roles would be small as otherwise the task of creating these sets of preferences would be too arduous.

In general the role-based approach has an advantage for personalisation as it reduces the complexity of the user preferences. This approach was implemented and demonstrated for several scenarios. However, it does have two main drawbacks:
(1) Establishing a set of user preferences for each role can become too arduous and the user will rapidly lose interest in doing so. This depends on the number of different roles (VIDs), each with its own independent set of preferences, as well as the number of different kinds of services that

the user makes use of. This task could be eased with the use of some form of learning (see section 5).

(2) Knowing the user's role at all times places an additional burden on the user. The only way in which we could see this being handled was by the user informing the system whenever he/she changed role. For some situations this might be straightforward – switch to work mode when the user arrives at work, switch to leisure mode when the user leaves work, switch to holiday mode when the user goes on holiday. However, there are other cases where work and leisure are more closely intertwined; for example, suppose that the user receives a phone call of a social nature while he/she is at work – does one switch to leisure mode for the duration of the call?

5. Handling privacy with non-role-based VIDs

In the second phase of the Daidalos project it has been decided to avoid the problems associated with role-based VIDs but instead allow the user to create an arbitrary set of VIDs that can be used how and when the user likes. Thus although the user may choose to allocate different VIDs according to different roles, he/she might equally allocate them completely randomly. Of course, in the simplest case he/she might only use a single VID, although this is not interesting as it offers minimal protection against invasion of privacy.

In addition, it is clear that expecting the user to set up and maintain a set of preference rules associated with the VIDs is unrealistic. Thus for the second phase of Daidalos it has been decided to provide additional support for the user in two stages. The first stage involves setting up a default set of preference rules for each user, possibly based on stereotypes, and the second will refine and extend these by monitoring the user's behaviour and using machine learning techniques on the data produced. Naturally the option must exist for the user to enter or change his/her own preference rules and to view the complete set of his/her preference rules at any time. But one should not rely on the user building this up on his/her own.

The complication with this approach already referred to in the previous section is that the system should not maintain any links between the different VIDs belonging to a user so that there is no trace that might allow some service to infer additional information about the real identity of the user. This assumption leads to a major drawback with regard to the process of learning user preferences. In order to build up a set of user preferences rapidly one needs to know if the same set of preferences are shared by a group of VIDs. Then whenever the user is accessing the same services, even with different VIDs, the same set of preferences can be used. Otherwise the learning process will have to be applied to each VID separately. This will not only take a lot longer to achieve but also lead to considerable user frustration as the system makes the same mistakes repeatedly for different VIDs until all are consistent – which, in general, may never happen as the user's preferences themselves will change with time.

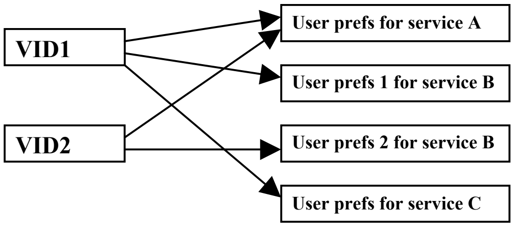

Fig. 1. VIDs 1 and 2 have common preferences for service A, different ones for B and only VID 1 has preferences for service C.

In order to overcome this problem a mechanism is needed to enable the user to be able to tell the system to share profile information between different VIDs. This may be achieved in different ways and is beyond the scope of this paper. The important thing is that the personalization processes have access to profiles that may be shared by several VIDs belonging to the same user, and that a secure access system is provided to protect from unauthorised access to these profiles.

The result is that one should allow sharing where this is required but otherwise maintain separate preference rules. For example, suppose that VID1 should have a common set of preferences with VID2 for service A but that they should have different preferences for service B and that service C is used only by VID1 and not by VID2. This is illustrated in Fig. 1.

6. Proactive behaviour

Another important issue in a pervasive system is the degree to which such a system should be pro-active. Generally, pro-activity is associated with user context, and may also be linked to some form of personalization. For example, if the user is in a particular location or performing a particular task, then perform some action to prepare for a subsequent task. An example of this might be if the user is working on a particular document, or set of Powerpoint slides for a lecture, the system might download a set of relevant files that the user might find helpful. Another commonly quoted example is that of a user for whom, when entering his/her sitting room and sitting down in his/her favourite chair, the system automatically switches on the television to his/her favourite channel.

In order to handle pro-activity it is useful to separate pro-active rules from the more passive personalization

associated with applying user preferences. Thus in the above example of the television set, one can divide the problem into a pro-active rule:
"If user X sits down on chair Y, start TV service"
and a user preference:
"If TV service invoked then preferred channel = C".

Such pro-active rules can be dealt with by a separate service such as a Personal Assistant service. Although some aspects of pro-activity may be handled by general rules that apply to all users, others will inevitably be user-specific. Once again learning can be used to maintain and refine the user-specific pro-active rules.

In handling pro-activity care must be taken to avoid further conflict with privacy. In general one can distinguish two types of situations:
(1) Pro-active behaviour triggered by others. A simple example of this might be a music shop that wants to alert users of a new release when they are close to the shop. Thus if a user has registered one of his/her VIDs with the music shop, when that user is in the vicinity of the shop, a message will be sent to him/her. However, the user may have a number of different VIDs, some for professional use, others for private use. If the user has registered with one of his/her private VIDs (say priVID1), then if he/she is currently in professional mode, the music shop should not be able to access his/her location. But, more importantly, how does the system know which VID the user is currently using if he/she is not actually executing some service. Beside, it is possible for the user to have more than one service operating at any instant, and hence potentially more than one VID in use.

Where the pro-active action is the initiation of some form of communication – voice call, voice message, SMS, email, etc. – this may be controlled by the rules for personalized call redirection mentioned in section 2. Using the user's current context it can block or redirect calls or messages so that the user's privacy is not invaded by ill-timed messages.

(2) Proactive behaviour triggered by the system. A similar kind of situation could arise with actions set up by the user, although here it extends beyond simple messaging. Consider the case where a user sets up the system to switch on the TV set to his/her favourite channel when he/she sits down in the sitting room. When this is set up it must be created for a particular VID. If the user wants it to hold for all of his/her VIDs, he/she must enter the same rule for each VID separately. Once again the problem arises as to which VID is operational at any one time and hence whether or not the action should be performed.

The problem is further complicated if one is trying to apply machine learning to predict proactive behaviour patterns.

7. Conclusion

Personalization has an important role to play in pervasive systems but clearly creates a problem with regard to privacy. This conflict has attracted research on privacy and trust in pervasive systems. However, Chatfield et al [17] have shown that users are willing to divulge personal information as long as they receive a substantial benefit in return. Chellappa and Sin (2005) surveyed 243 people to arrive at a model in which the value of personalization, the likelihood of using a personalization service, concern for privacy and the existence of trust building factors in a system were causally linked [18]. A useful survey conducted on 4520 users, by the new Personalization Consortium, showed that users want both personalization and privacy and are willing to share personal information as long as their privacy is protected [19].

This paper outlines some types of personalization that can be used in a pervasive system to provide services tailored to individual user needs. The issue of privacy in a pervasive system is discussed and the three main areas that need to be protected are identified. Of these the paper focuses on the protection of information on the services run by the user.

The paper then discusses two slightly different approaches adopted in the Daidalos project. The first of these uses role-based virtual identifiers (VIDs) and has been implemented in the first phase of the project. The second approach is based on multiple VIDs with no particular relationship associated with them and will be implemented in the second phase. These two approaches can provide adequate protection of the information on the services run by the user although they have drawbacks for personalisation, especially with respect to building up sets of user preferences.

Pro-activity is another important aspect of pervasive systems and one approach to dealing with it was outlined. A potential problem with pro-activity and the use of multiple VIDs was identified. By use of appropriate personalization techniques (personalized call redirection) this problem can be minimized.

Acknowledgements

This work was supported in part by the European Union under the FP6 programme (Daidalos project) which the authors gratefully acknowledge. The authors also wish to thank all colleagues in the Daidalos project developing the pervasive system. However, it should be noted that this paper expresses the authors' personal views, which are not necessarily those of the Daidalos consortium. Apart from funding the Daidalos project, the European Commission has no responsibility for the content of this paper.

References

[1] M. Satyanarayanan, Pervasive computing: vision and challenges, *IEEE PCM, 8*(4), 2001, 10-17.

[2] A. Zaslavsky, Adaptability and interfaces: key to efficient pervasive computing, *Proc. NSF Workshop on Context-Aware Mobile Database Management*, Providence, Rhode Island, 2002, 24-25.

[3] C. Gbaguidi, J. P. Hubaux, et al., An architecture for the integration of Internet and telecommunication services, *Proc. IEEE Infocom '99*, New York, Mar. 1999.

[4] N. Faggion & C. T. Hua, Personal communications services through the evolution of fixed and mobile communications and the intelligent network concept, *IEEE Network*, Jul./Aug. 1998.

[5] B. Farshchian, J. Zoric, L. Mehrmann, A. Cawsey, H. Williams, P. Robertson and C. Hauser, Developing Pervasive Services for Future Telecommunication Networks, *Proc. WWW/Internet 2004*, Madrid, Spain, October 2004, 977-982.

[6] D.Riecken, Personalized views of personalization, *Comm. ACM, 43*(8), Aug. 2000, 26-28.

[7] D. Bental, L. MacKinnon, H. Williams, D. Marwick, D. Pacey, E. Dempster & A. Cawsey, Dynamic Information Presentation through Web-based Personalisation and Adaptation - an Initial Review, *Proc. HCI2001 and IHM2001*, 2001, 485-500.

[8] IBM, Tivoli Personalized Services Manager, V. 1.2. ftp://ftp.software.ibm.com/software/pervasive/info/tech/tpsm_ss.pdf, 2002.

[9] A. Brar, and J. Kay, Privacy and Security in Ubiquitous Personalized Applications, *Proc. User Modelling Workshop on Privacy-Enhanced Personalization*, Edinburgh, UK, 25 July, 2005.

[10] Q. Z. Sheng, B. Benatallah, et al, Enabling Personalized Composition and Adaptive Provisioning of Web Services, *Proc. 16th Int. Conf. on Advanced Information Systems Engineering (CAiSE)*, Riga, Latvia, June 7-11, 2004.

[11] F. Casati, S. Ilnicki, L. Jin, et al, Adaptive and Dynamic Service Composition in eFlow, HP Labs 2000 Technical Reports, www.hpl.hp.com/techreports/2000/

[12] Project Aura Home Page, Carnegie Melon University [cited 24th January 2007] http://www.cs.cmu.edu/~aura/

[13] J. P. Sousa, V. Poladian, D. Garlan, B. Schmerl & M. Shaw, Task-based adaptation for ubiquitous computing, *IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. 36*(3), May 2006, 328-340.

[14] N. Hanssens, A. Kulkarni, R. Tuchinda & T. Horton, Building Agent-Based Intelligent Workspaces, *Proc ABA Conference*. June 2002.

[15] Y. Yang, F. Mahon, M. H. Williams & T. Pfeifer, Context-aware Dynamic Personalized Service Re-composition in a Pervasive Service Environment, *Proc. 3rd IFIP Int. Conf. on Ubiquitous Intelligence and Computing (UIC 06)*, Wuhan, China, Springer Verlag LNCS 4159, 2006, 724-735.

[16] M. H. Williams, Y. Yang, L. MacKinnon, R. Dewar & N. Milyaev, Personalized Redirection of Communication in a Pervasive System, *Proc. 12th Int. Conf. on Telecommunications*, Cape Town, 2005.

[17] C. Chatfield, D. Carmichael, R. Hexel, J. Kay & B. Kummerfeld, Personalisation in Intelligent Environments: Managing the Information Flow*, Proc. OZCHI*, Canberra Australia, 2005.

[18] R.K. Chellappa & R.G. Sin, Personalization versus Privacy: An Empirical Examination of the Online Consumer's Dilemma, *Information Technology and Management, 6,* 2005, 181-202.

[19] Personalization Consortium, Wanting it all: Give us Personalization AND Privacy [online]. April 2000 [cited 24th January 2007]. HTML. Available from: http://www.1to1media.com/view.aspx?DocID=12396 .