Context-aware user preferences in systems for pervasive computing and social networking

Elizabeth Papadopoulou, Sarah Gallacher, Nick K. Taylor, M. Howard Williams and Fraser Blackmun

School of Math & Computer Sciences, Heriot-Watt University,
Edinburgh EH14 4AS, UK
{E.Papadopoulou, S.Gallacher, N.K.Taylor, M.H.Williams, F.R.Blackmun}@hw.ac.uk

Abstract. Context-aware user preferences play an important role in adapting the behaviour of pervasive systems to satisfy the individual user in different contexts. Most of the pervasive system prototypes that have been developed, incorporate context-aware user preferences, capturing them in an appropriate form and using them to personalize the behaviour of the prototype. However, the rapid rise in use of social networking systems has created a new possibility – that of combining this type of system with pervasive systems to create a new type of system with the benefits of both. The challenge lies in finding how to integrate these two different paradigms in a seamless way to create such a system. This introduces new challenges for the context-dependent user preferences for both individuals and communities. This paper describes briefly some of the ideas, especially as they relate to context-aware user preferences.

Keywords: Context-aware, user preferences, learning, personalization, pervasive systems, social networking.

1 Introduction

Pervasive computing and social networking are two separate paradigms that are of great importance to current and future systems.

The first of these, pervasive computing, is concerned with the problems arising from the rapid growth in the number of devices in the environment surrounding the user, and the overwhelming number of services available to her. A major aim of pervasive computing is to provide the essential support that a user needs to help her communicate with and interact with this increasingly complex environment. This includes interaction with and control of devices and services while protecting her from the underlying complexity in doing so [1]. Many prototypes have been produced to experiment with the ideas of pervasive computing (e.g. Adaptive House [2], Ubisec [3], GAIA [4], MavHome [5], Things That Think (TTT) [6], Daidalos [7]).

In order to be acceptable to the end-user, a pervasive system must be able to adapt its behaviour to take account of the needs and preferences of individual users. Since the user cannot be expected to provide the system with the information to do so, the

approach generally adopted is for systems to monitor the user's behaviour and infer the user preferences from it. However, this task is both complex and challenging. Indeed one of the major challenges facing the developers of pervasive systems is to build up a subset of user preferences which, although incomplete, is sufficiently useful that it can provide a level of adaptive behaviour that is acceptable to the user.

An example of a pervasive system platform that operates in this way is that developed in the Persist project [8]. This was based on the use of Personal Smart Spaces for the development of pervasive systems. The prototype platform developed is founded on the notions of context-awareness and of personalisation based on user monitoring and learning to establish and refine context-dependent user preferences. The prototype has been used to demonstrate the ideas and to experiment with them.

The second paradigm, social networking, is one that has been hugely successful. Facebook has become a household name and systems such as LinkedIn, YouTube, Flickr, Skype, etc. have all become very popular with a very large user base. Social networking itself has been advancing rapidly and the latest developments in this area involve the incorporation of user context into such systems. The main focus has been on user location, and systems such as FourSquare rely entirely on this.

However, a new challenge lies in bringing together these two different paradigms, social networking and pervasive computing, in a fully integrated fashion. This would mean that the user could interact with friends and contacts via the social networking functionality as well as with services and devices in her environment through the pervasive functionality in a seamless manner. This has particular advantages for context and personalization which can be used to support both.

This paper describes some of our current research in this area. The next section describes some aspects of the Persist pervasive system platform while section 3 discusses briefly the issue of combining social networking and pervasive capabilities, especially with respect to context-dependent user preferences. Section 4 concludes on the state of implementation.

2 Pervasive Computing and Persist

The approach developed in the Persist project for handling pervasive behaviour is based on the notion of a Personal Smart Space (PSS). A PSS consists of a collection of devices belonging to a single owner (user or organisation) that are connected to form an ad hoc network. This may be mobile (in the case of a person who may move around with her PSS) or fixed (in the case of a fixed smart space, e.g. a smart home).

When one PSS encounters one or more other PSSs, they may interact with one another. This involves a PSS identifying itself to the other PSSs, subject to the constraints of user privacy. They may then proceed to share information or even third party services with other PSSs. This is described in more detail in [8].

Using the concept of Personal Smart Spaces the Persist project built a pervasive system prototype to demonstrate some of the capabilities that this can provide. This was based on an architecture consisting of five layers, as illustrated in Fig. 1.

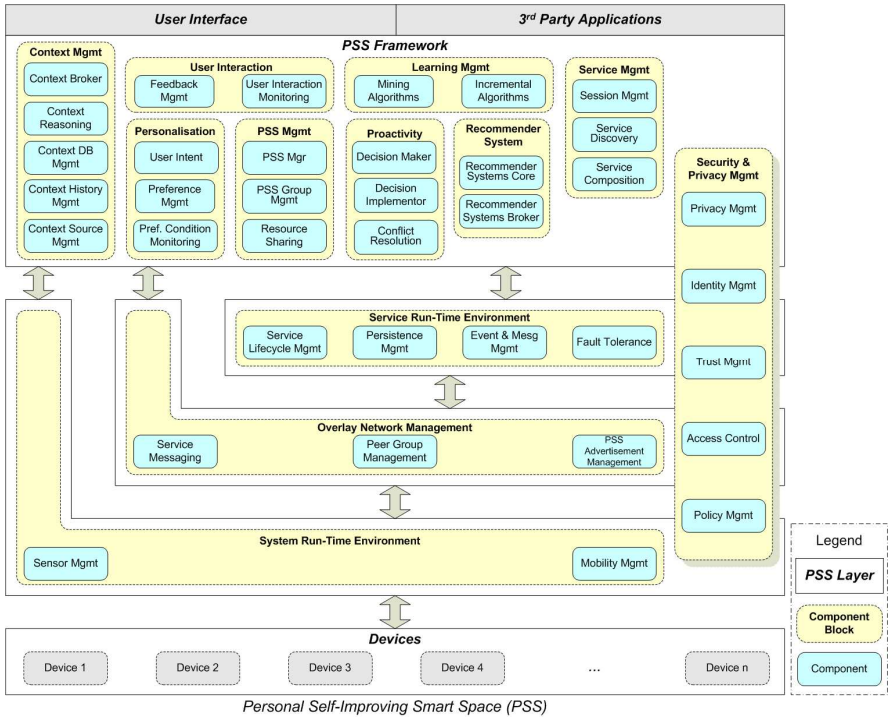

Fig. 1. The high level architecture of a Personal Smart Space

Central to this system is the notion of context-aware personalization. By this we mean the adaptation of the behaviour of a system to meet the needs and preferences of an individual user. This is an essential feature of the Persist platform. To achieve this two separate mechanisms are used. The first of these is based on user preferences. Each user has a set of user preferences associated with them which are used to tailor system behaviour to meet these preferences. In particular this is used to tailor individual services on behalf of the user. This functionality is provided by the Preference Manager and the Preference Condition Monitor in the Personalisation subsystem.

The second mechanism that is used in Persist is based on models of User Intent. In this case models are constructed over time of the sequences of actions performed by the user and typical recurring sequences are identified. These are then used to predict future actions that the user may wish to take. This is handled by the User Intent component of the Personalisation subsystem.

These two mechanisms predict actions which the system should take in order to personalize its behaviour in a way that might help the user. However, overall control of such actions is handled by the Proactivity subsystem which takes the final decision as to whether an action should be performed or not, and resolves any conflicts that may arise between the actions proposed by these two mechanisms.

Learning also has a key role to play in the Persist system. In order to capture these user preference rules, the Persist platform monitors the actions taken by the user and

stores these together with the state of selected context attributes at the time of the action. Two forms of machine learning are then used to extract and refine user preferences. The one is an offline datamining algorithm which is used to analyse the whole data set. This is slow and only used at convenient times (e.g. at night). The other is an incremental algorithm which is applied only to the latest subset of the data. This is much faster and is applied when needed (e.g. when the actions predicted by the preference rules lead to a conflict). The overall process is illustrated in Fig. 2.

Although the main elements of context used in the preference rules were the symbolic location of the user and some categorization of time/day, we also experimented with other sensors, including ones that determined the user's current state in terms of sitting, standing, walking, etc.

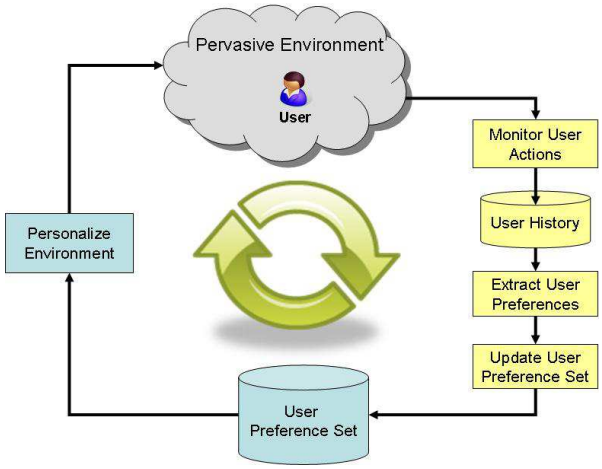

Fig. 2. Process of building and updating the set of context-dependent user preferences

As can be seen from Fig. 1, the issues of Privacy and Security are treated very seriously in the Persist system. In the case of the context-dependent user preferences, three strategies are employed to protect the information on the user, viz.

(1) The user preference rules are held by the Preference Management component of the system, which provides two routes to them. The first is a GUI which the user can access to view and, if necessary, change preferences. The second is the route provided to all other system and third party software which wants to use preferences. In the latter case the software calls the Preference Manager, which evaluates the appropriate rules whenever required, returning any action that results from this evaluation. The component cannot return the preference rule itself. Thus no component can obtain unauthorized access to this information through this route.

(2) In addition to this basic protection mechanism, the Persist system also supports multiple identities. Thus a user can select different identities to use with different services. Once again user preferences are used to assist the user in the process of selecting identifiers. This has the added advantage that this selection process is context-aware. Thus the system may select different identities for the same third party

service in different contexts. This helps to protect the real identity of a user when using different services.

(3) The final protection mechanism investigated for the Persist platform involves the use of privacy policies. Here the user must specify a set of privacy policy rules that specify what information can be given to what services and under what conditions.

To assess the effectiveness of our approach to context-aware user preferences, an experiment was conducted to learn viewing preferences for television. This was based on the fact that viewing preferences at home might be different from those at work. Twenty four volunteers took part and used the prototype to select television channels to watch on various plasma screens at different locations in our building. After an initial learning phase, the participants were told to change some of their selections and repeat the exercise. Results show an accuracy of 100% after two circuits of the initial learning phase. After changing selections the system took up to five circuits to learn the changes. The overall response to the platform was very positive.

3 Combining Pervasive and Social

In order to combine pervasive computing with social networking we have made two major changes. The first is to extend the basic ideas of pervasive computing to incorporate both individuals and communities. The idea of a community can be a fairly complex one. A community may have its own criteria in terms of membership, it may specify the types of information that members are prepared to share with each other and the set of third party services to which members may have access, it may be created manually or automatically by the system, and so on.

The second major change is to the PSS. Where the PSS was created to handle pervasive systems, a more general concept is needed to handle the combination of pervasive computing with social networking. For this purpose the concept of PSS has been extended to that of a Cooperating Smart Space (CSS) which has some of the properties of a PSS but also includes functionality for social networking. This is described in detail in [10].

The most important extension affecting context-aware preferences introduced in the CSS is the notion of community preferences. Just as individual users may have preferences associated with them, communities too may have preferences. For example, suppose that one creates a community of first year Computer Science students at a university. This community may have a particular preference for where to meet for lunch on a weekday at university, or for getting together on a Wednesday afternoon to play/watch football.

The main challenge here is how to derive these community preferences. On the one hand they could be obtained by extracting the individual preferences from the preference sets of each of its members and analyzing them to look for clusters. On the other hand, one may aggregate the history data from all its members and analyze this to extract preferences that apply to the whole group.

Another mechanism for obtaining preferences is by inheritance. Within any community one may also have sub-communities – subsets of the membership of the parent community who are linked together for some purpose. A sub-community may inherit preferences from the parent community as well as having its own unique preferences.

More generally, the preferences of individual users will be updated whenever the system discovers new preferences or changes to existing ones. The same applies to the management of community preferences. One major problem is that of how to deal with conflicts – such as those that may arise when the individual user preference results in a different outcome from that of a community preference.

Another important issue relating to communities which needs to be taken into account concerns how these are created. One obvious way of creating a community is for a user to set this up. However, a more challenging problem is for the system to identify potential communities dynamically. This involves analyzing the data relating to a set of CSSs and looking for clusters based on their attributes.

To illustrate the power of such a system, consider the following extract from the set of scenarios used to illustrate the behaviour of such a system:

Scene 1: Harry is a new student who has just arrived at Heriot-Watt University (HWU). He is alerted to important communities that he is strongly encouraged to join, particularly the "Freshers" community that all new students can join. On joining the Freshers community Harry inherits several community preferences. One such preference is the preferred venue to buy lunch on campus. He is also automatically added to a "Computer Science" community for the degree course he is taking.

Scene 2: That evening Harry attends a Freshers' event called the "Proactive Disco". It is a community based disco that takes into account the individual user preferences relating to music of all the people currently dancing on the dance floor (identified using sensor technology) and decides what music tracks to play.

Scene 3: Harry is leaving his dorm room to attend his first lecture. His CSS identifies his intent to attend the lecture and the navigation service is automatically started to direct Harry to the lecture room. On his way Harry's CSS identifies another person nearby who also has the intent of attending the same lecture. Since they share intents and since the other person's mood is 'happy', Harry's CSS suggests an introduction which he accepts. Harry's CSS shows him a picture of the other person and tells Harry his name is Tom. They begin to chat as they walk together to the lecture.

4 Implementing a Pervasive Social Networking System

The set of requirements that has emerged for this type of system is much more extensive than that for a pervasive system such as the Persist system. The Societies project [9] is building on the work done in Persist to create a platform that combines pervasive computing and social networking in a seamless way, and which exhibits both types of behaviour. This is based on a rather different architecture that can deal with this combination of functionalities. For example, the issue of scalability takes on new significance when communities can end up with thousands of users. Similarly,

the way in which context-dependent user preferences are handled becomes more complex – with strategies employed to learn both individual and community preferences and to arbitrate between them when necessary.

To assess user reaction to these ideas, we conducted two evaluations. In the first a group of volunteers was given a storyboard presentation illustrating the behaviour of the platform and questioned on their reactions to aspects of the system.

In the second a physical experiment was set up with a version of the platform so that volunteers could experience the type of behaviour at first hand, and once again they were questioned on their reactions.

The conclusions from this exercise relevant to this paper were:

(a) The participants found the idea of the system taking decisions on their behalf more acceptable after experiencing the system than before it. In answer to the question "Would you allow the system to take decisions on your behalf …" 47% responded positively before the experiment while 77% responded positively after it.

(b) Asked about whether they would mind if the system monitored them in order to obtain improved user preferences, only 7% responded negatively.

The first prototype of the Societies platform is nearing completion. It is based on Android phones linking to the cloud for the main processing and storage requirements. The system will be exposed to different types of users and its performance evaluated in three separate user trials in the last quarter of 2012/first quarter of 2013. These are:

(1) Student trial, in which a number of students will be given the platform to use over an extended period.

(2) Disaster management trial, in which the system will be employed by real disaster management end users.

(3) Enterprise trial, in which the system will be used by industrial users for typical situations in commerce and industry, including conference type applications.

More details of this can be found at http://wiki.ict-societies.eu.

Acknowledgement

Besides thanking the European Commission for their support for the PERSIST and SOCIETIES projects, the authors also wish to thank colleagues in the two projects without whom this paper would not have been possible. Apart from funding these two projects, the European Commission has no responsibility for the content of this paper.

References

1. Satyanarayanan, M.: Pervasive computing: vision and challenges, IEEE PCM, 8(4), 10-17 (2001)
2. Mozer, M. C.: Lessons from an Adaptive House. In: Smart Environments: Technologies, protocols and applications, Cook, D. and Das, R. Eds., pp. 273-294 (2004).

3. Groppe, J., Mueller, W.: Profile Management Technology for Smart Customizations in Private Home Applications. In: 16th Int. Workshop on Database and Expert Systems Applications (DEXA ’05), pp. 226-230 (2005).
4. Kindberg, T., Barton, J.: A web-based nomadic computing system, Computer Networks, 35, 443–456 (2001).
5. Youngblood, M.G., Holder, L.B., Cook, D.J.: Managing Adaptive Versatile Environments. In 3rd IEEE Int. Conf on Pervasive Computing and Communications (PerCom '05), pp. 351-360 (2005).
6. Mistry, P., Maes, P.: Sixth Sense: A Wearable Gestural Interface. In: International Conference on Computer Graphics and Interactive Techniques, Yokohama, Article 11 (2009).
7. Williams, M. H., Taylor, N. K., Roussaki, I., Robertson, P., Farshchian, B., Doolin, K.: Developing a Pervasive System for a Mobile Environment. In: eChallenges 2006 – Exploiting the Knowledge Economy, IOS Press, pp. 1695 – 1702 (2006).
8. Crotty, M., Taylor, N., Williams, H., Frank, K., Roussaki, I., Roddy, M.: A Pervasive Environment Based on Personal Self-Improving Smart Spaces. In: Workshop on Constructing Ambient Intelligence, AmI 2008, Springer Verlag CCIS, vol. 32, pp. 58-62 (2010).
9. Gallacher, S., Papadopoulou, E., Taylor, N.K., Blackmun, F.R., Williams, M.H., Roussaki, I., Kalatzis, N., Liampotis, N., Zhang, D.: Personalisation in a System Combining Pervasiveness and Social Networking. In: 1st Workshop on Social Interactive Media Networking and Applications (SIMNA 2011), ICCCN 20, Hawaii, (2011).
10. Gallacher, S., Papadopoulou, E., Taylor, N. K., Blackmun, F. R., Williams, M. H.: Intelligent Systems that Combine Pervasive Computing and Social Networking, Proc. Ninth International Conference on Ubiquitous Intelligence and Computing (IEEE UIC 2012), IEEE Computer Society, pp. 151-158 (2012).