Intelligent systems for pervasive computing and social networking

Elizabeth Papadopoulou, Sarah Gallacher, Fraser Blackmun, Nick K. Taylor, and M. Howard Williams
School of Math & Computer Sciences
Heriot-Watt University
Edinburgh, UK
{ E.Papadopoulou, S.Gallacher, F.R.Blackmun, N.K.Taylor, M.H.Williams}@hw.ac.uk

Abstract—Social networking systems and pervasive computing are two essential paradigms for systems for the future. The former are well established and in use daily by most users. The latter are equally important although the ideas here have not had the same take up. The aim of the Societies project is to integrate the two concepts in a seamless way to create a new type of system with the benefits of both.

Keywords—pervasive; social networking; community.

I. INTRODUCTION

Pervasive computing is aimed at providing the essential support that a user needs to help her communicate with and interact with the environment around her. This includes interaction with and control of the growing number of devices surrounding her and managing the huge array of services to which she has access. Many prototypes have been produced to experiment with the ideas of pervasive computing (e.g. Adaptive House [1], Ubisec [2], GAIA [3], Daidalos [4]) although none have resulted in commercial take up.

On the other hand, social networking is a paradigm that has been hugely successful. Facebook has become a household word and systems such as LinkedIn, YouTube, Flickr, Skype, etc. have all become very popular with a very large user base.

The aim of the Societies project is to combine these two paradigms and create a new type of system that exhibits both pervasive and social networking functionality, integrated together in a seamless fashion.

II. PERVERSIVE COMPUTING

The approach being used to handle pervasive behaviour is based on that used in the pervasive prototype developed in the Persist project [5]. In this system the notion of a Personal Smart Space (PSS) is introduced to denote 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, such as 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.

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.

Central to this system is the notion of personalization (or context-aware personalized adaptation). By this we mean adaptation of the behaviour of a system to meet the needs and preferences of an individual user. This is an essential feature of any pervasive system. Each user will generally have a set of user preferences associated with them which are used to tailor system behaviour to meet these preferences. This includes tailoring of individual services as well as taking pro-active actions on behalf of the user. This is where the intelligent behaviour of the system resides.

In order to capture these user preferences and pro-active rules, some form of automatic learning is generally provided to assist the user. In the Persist project three different approaches were used to represent user preferences (if-then rules, neural nets and Bayesian networks) and several different learning strategies were incorporated.

One of the problems encountered in pervasive systems such as Persist is the identification of what information and which services to share with another PSS. For example, depending on the relationship between one PSS and another (e.g. if the other PSS is a friend, a relation, a work colleague, a client, etc.), the user may have different criteria regarding what can be shared and with whom. One way of assisting in this process is for users to identify groups of PSSs, which can be used to help in deciding what information or services can be shared with any particular PSS. For example, a PSS could specify groups of users in different categories who can be given access to particular items of information (such as location) or particular services. In other words this can be used to help in determining the level of trust associated with a PSS.

III. COMBining pervEsive and social

In seeking how to combine pervasive functionality with social networking, the idea of a Cooperating Smart Space (CSS) is introduced. This has some of the properties of a PSS but also includes functionality for social networking.

In addition, the concept of a community is introduced. This is far more complex than the idea of a group inPersist since it is not used simply to determine access to information and services but as a more general concept. For example, a community may have its own criteria in terms of membership, including the types of information that
members are prepared to share with each other. A community may also have associated with it the set of third party services that members may have access to.

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. These community preferences could be obtained by extracting the individual preferences from the preference sets of each of its members and analyzing them to look for clusters. Alternatively, one may aggregate the history data from all its members and analyse this to extract preferences that apply to the whole group.

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 that needs to be addressed 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 currently being used to drive the development of the Societies 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 music preferences 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 flags 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.

The set of requirements that is emerging for this type of system is much more extensive than that for the Persist system and a new type of architecture is called for 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, tens of thousands or even greater numbers of users.

Thus, in order to create the type of functionality in the platform that extends beyond the individual to dynamic communities of users, the Personalisation system needs to address additional requirements, which entail new research challenges. The Societies project aims to produce such a system. The platform that is being developed will be exposed to different types of users and its performance evaluated in three separate user trials. 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](http://wiki.ict-societies.eu).

ACKNOWLEDGMENT

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


