Towards An Infrastructure for Pervasive Recommendations

Paul Dourish

Computer Science Laboratory
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto CA 94304 USA
dourish@parc.xerox.com
http://www.parc.xerox.com/dourish

INTRODUCTION

The most common form of social navigation, and the form in which it has become most familiar to people, is through recommendation systems. Recommendation systems help people navigate through an information space, typically one of homogeneous items of information, by deriving interest profiles for each user and comparing them to each other, so as to be able to specific options to one user by finding items that have been of interest to users with similar profiles.

This approach has been successfully used with items such as books, movies and CDs, which has in turn meant that recommendation systems have been popular with e-commerce web sites, for two reasons. First, it encourages people to buy more items by allowing the system to bring to their attention potential purchases that they might otherwise have overlooked; and second, it builds "buy-in" by encouraging users to stick with one service so as to build up enough of a "purchase history" that the system can tune itself to an individual consumer's preferences.

Although this has proved to be a valuable and successful way to harness social navigation in interactive systems, it is none the less a limited approach. A variety of systems have explored alternative systems that can incorporate the use of social navigation and recommendation into other sorts of applications. In particular, they have explored how social navigation can be used to augment either special-purpose applications or traditionally single-user applications such as email and news readers, so that an individual user can capitalise on the activities of others in accomplishing their work.

Although these approaches go extend the notion of recommendation into our interaction with systems, rather than simply the recommendation of commodity items, they still restrict social navigation to being a feature of specific applications. In contrast, in this paper, I want to explore a notion I call "pervasive recommendations". I will outline what I mean by this term, discuss some of the requirements for an infrastructure to support it, and explore some of the issues that it might open up.

PERVASIVE RECOMMENDATIONS

Pervasive recommendations refers to the idea that the use of recommendations – hints for action based on similarity profiling over a community – as a basic element of interactive infrastructure. Pervasive recommendations are available to me throughout my interaction with the software systems, not simply in specific applications, and certainly beyond the scope that we might traditionally associate with recommendations. One example, which I owe to Paul Resnick, is workstation network settings – the TCP/IP Control Panel under MacOS, or the TCP/IP Properties pane under Windows. This is exactly the sort of information that, first, generally is not worth the time investment required to just get it working; second, will often involve calling on others; and third, can be usefully approximated from available information about others. My network mask will almost always be the same as that of the person in the office next door; my IP address will generally start with the same few digits.

Being able to infuse all aspects of the interactive experience, right down to the TCP/IP control panel, with information gleaned from the activities of other users is the heart of the pervasive recommendation vision. It operates not on the level of individual applications, whether those are specialised community applications or augmented single-user applications; rather, it operates throughout the system, "pervasively."

A MODEL FOR PERVASIVE RECOMMENDATIONS

When we begin to think of extending social navigation through out the system, we encounter a number of challenges to the way in which conventional social navigation systems operate. The most significant of these is the set of design parameters. Traditional social navigation systems recommend only one sort of item (e.g. books), or where they recommend multiple types (e.g. books and CDs), they operate in largely the same ways.[1] Supporting pervasive recommendations, however, requires that we make explicit various design parameters that are otherwise left implicit in specific-application approaches.

For example, take the "similarity community" – that is, the set of people whose activities may be relevant to my own. For purchases, that is built on the basis of people's purchase profiles. For email recommendations, it may be people who

1. There is some debate over the effectiveness of "cross-modal" recommendations (of the form, "if you liked those books, you'll probably like these CDs"); but independently of that issue, books and CDs are both recommended in the same sorts of ways, using the same sorts of purchase profiles over the same community of people.

subscribe to the same mailing lists as I do – not just "similar" lists, since we want recommendations about specific messages. For certain forms of information recommendation, the similarity community might be defined organisationally (so, for me, it would constitute people in the same workgroup, or in the same lab or center). For TCP/IP control settings, though, it will be defined geographically; I want to know the settings of other people close to me, since networks are typically managed geographically. For pervasive recommendations to work – that is, for a single infrastructure to be able to support a wide range of different sorts of recommendations in different circumstances – features like this must become explicit parts of the model.

Exploring the components of the model reveals something roughly like this. The following functions are defined:

- For any application A[2], the set of community members R_A whose actions are relevant to the actions of an individual X.
- For those members, the set V_R of application values (books, network settings, etc.)
- For those same members, the set W_R of relevance weightings
- From W_R and V_R, a set WV of weighted values;
- A function G over WV that generates a set of generalised values. This may be calculated by simply selecting a group of elements form the value space (e.g. a set of books), or by generalising over the data values themselves (e.g. finding the common elements in a set of IP addresses).

By going through these steps, we can derive a set of suggestions or recommendations for the application's settings according to the behaviour of others.

INFRASTRUCTURE

Any recommendation system includes the steps outlined above, although they may often be encoded implicitly in the design of the software systems itself. Pervasive recommendations requires, though, that the definitions of these stages be separated from the code so that different definitions can be active concurrently for the different applications that might be in operation. The core requirement for an infrastructure for pervasive recommendations is that it operates independently of application and, therefore, across multiple applications simultaneously.

Based on the model presented above, then, the infrastructure has three functions. First, it collects and collates information about application parameter settings and data items. Second, it maintains a variety of functions over this information for different applications – different selection and weighting functions, different generalisations, and so forth. Third, it provides specific recommendations in response to requests from per-user application instances.

2. Although I use the term "application" here, my focus is not on "applications" as much as domains. So, for instance, I take TCP/IP settings to be an application, even though the settings themselves might be system variables rather than application entities.

The design of such an infrastructure is being explored, based on a loosely-structured information system called "Placeless Documents" [1]. Amongst other things, Placeless provides an associative data store that is shared, persistent, extensible and transparently distributed.

Anonymity and Identity

One interesting issue occurs with the pervasive recommendation approach, which is the question of anonymity and "warranted recommendations." This issue is highlighted by studies such as those of Mackay [2] into patterns of sharing customisations in extensible software systems. Mackay traced the history of customisations and investigated their trajectories. What she found was that local experts often emerged as the source of customisations for a specific work group, or as the gatekeeper through whom customisations enter local work groups. These people are not only mediators; they are also trusted by local workgroup members, and they "warrant" or legitimate the customisations ("I can trust this .login file because it comes from Jim.").

Amazon.com recently introduced some changes to their recommendation mechanisms that highlight this issue. Standard recommendations, based on your purchase history, are naturally anonymised -- they are the result of putting together information from many people, and so come from no one person in particular. On the other hand, written reviewers do come from particular individuals. Their interface now allows people to rate reviews, so that others, later, can decide how much faith to put in the reviews. These changes reflect the different concerns in different sorts of information – sometimes anonymised information is acceptable, but sometimes I want to know what particular people think, because I trust specific individuals. Similarly, in some circumstances – such as TCP/IP settings – I might be happy to use information generated across a large number of people, such as everyone in my work site. In other cases, such as user interface configurations, I may want to rely on the information of a specific individual. This is especially in the case where there are finer gradations in the information (more or less correct, rather than working or not), where there are issues of coordination (one system's settings affect those of another), or where there are matters of individual taste at work.

CONCLUSIONS

The essence of this proposal is that social navigation should be seen as an infrastructure feature rather than a feature to be made available on an application-by-application basis. The hope is that, by providing a flexible infrastructure in which the selection and generalisation of information are pluggable, recommendation can be made a pervasive feature of interaction, so that even the most staunchly single-user applications can be made responsive to patterns of use within a community.

REFERENCES

1. Paul Dourish, Keith Edwards, Anthony LaMarca, John Lamping, Karin Petersen Michael Salisbury, Doug Terry and James Thornton. Extending Document Management Systems with Per-User Active Properties. ACM Trans. Information Systems. In Press.
2. Wendy Mackay. 1990. Patterns of Sharing Customizable Software. Proc. ACM Conf. Computer-Supported Cooperative Work CSCW'90 (Los Angeles, CA). New York: ACM.