Context-Aware Personalised Service Delivery

E Pignotti and **P Edwards** and **G A Grimnes** [1]

Abstract. In this paper we explore the potential of adaptive recommendation techniques in an environment where users can access a variety of services from different locations, and present a system (RECO) which implements these approaches.

1 Introduction

In this paper we discuss a system which employs a number of adaptive recommendation strategies [4] in an environment where users can access services. We argue that recommendations can only be made with a full understanding of context defined in terms of: date and time, user's location (home, office, ...), the history of services visited by the user, and service specific user preferences (user profile). The choice of profile mechanism to use in RECO was influenced by our previous work on an agent-based multi-service environment GraniteNights [1].

RECO (see Figure 1) attempts to build a model of its user by identifying patterns of service invocation, taking both the invocation sequence and the context into consideration. The *AprioriAll* [8] algorithm was selected to identify sequential patterns in user behaviour such as: `RESTAURANTS -> CINEMAS` (indicating that when the user visited the Restaurant service he also subsequently visited the Cinema service). Proximity rules determine if the user's location and current time are appropriate to receive a recommendation. The NNGE [5] algorithm was selected to learn these as it gave the best results when evaluated against a set of artificial training data [3]. Example proximity rules generated by the system are shown in Figure 3.

[1] Department of Computing Science, King's College, University of Aberdeen, Aberdeen, AB24 3UE, Scotland. Email: {epignott, pedwards, ggrimnes}@csd.abdn.ac.uk.

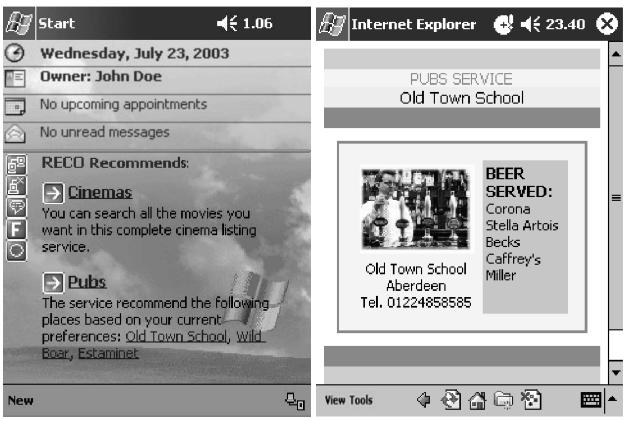

Figure 1. The RECO Client Application for Microsoft Pocket PC.

In a community of users it is possible to identify two kinds of service invocation pattern: one represents the behaviour of a single user in the system; the other represents the behaviour of the entire user community. Knowledge about the community of users can be used to generate recommendations for a user new to the system. Ultimately, context is managed within RECO by monitoring user behaviour, i.e. what services are invoked, in what sequence, where from and at what time.

2 RECO Architecture Overview

RECO is designed to recommend context-aware services to the user based on an existing user model. The model comprises the sequence rules and proximity rules generated by RECO, plus an existing RDF user profile [1]. The first recommendation technique analyses the user's behaviour in the environment and predicts the next service that he will invoke; the second utilises the user's preference profile to recommend more specific information.

At the core of the RECO system is a multi-agent architecture which provides the agents that are needed in order to monitor user behaviour and make recommendations. In order to communicate all RECO agents use standard FIPA(www.fipa.org) compliant messages envelopes, with content expressed in XML. The main agent roles are:

Environment Agent : Keeps a list of users and services currently connected to the system, and monitors changes in the environment when new users register or users access a service. The Environment Agent holds a "log" of all the services that a user visits, and continuously applies the AprioriAll and NNGE algorithms to the log in order to identify a model of user behaviour. This model is used to predict if a user will access a service in the near future, given the current time and location; such a prediction is used to alert the appropriate service agent which will then attempt to recommend a personalised version of the service content to the user. The Service Predictor and Alert System (Figure 2) is a module within the Environment Agent. Its task is to predict the next service that a user will invoke; it also must check if the time and user's location are appropriate for that recommendation.

The Environment Agent applies AprioriAll to the service invocation history of all users as well as to the history of each user independently, identifying a set of environment sequence rules as well as user specific ones as shown in Figure 3. The two sets of sequence rules are then used to predict the next service that a user will access; the context for this prediction is provided by the services already visited by the user during the current session. The system identifies rules that can be applied and uses the one with the highest support and confidence values to suggest the service that the user may next access. When the Environment Agent extracts the applicable rules it gives priority to the user sequence rules. If a user rule matches the

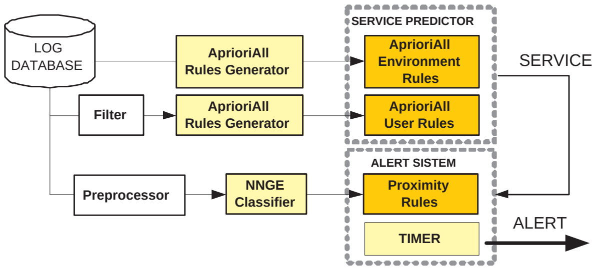

Figure 2. Service Predictor & Alert System - Environment Agent.

current user session, the environment service sequence rules are ignored. However, environment sequence rules can be used if a user has never explored certain services or if the user is using the system for the first time.

As explained above, the system aims to recommend services to the user at the right time and location. Sequence rules alone are insufficient, because time and location are not considered when they are generated. In order to overcome this problem a set of proximity rules are also generated from the log data using the NNGE algorithm. Figure 3 presents some examples of proximity rules generated by RECO. If the Service Predictor predicts the Cinema service as the next service, the first proximity rule in Figure 3 is used against the session information; if the user is at home, the day is Friday, Saturday or Sunday and the time is 17:25 the rule supports the service prediction. In this case, an alert message is then sent to the appropriate service. If the proximity rule is not valid for the current situation the Environment Agent continues to monitor the user until time and location are correct; during this period the Service Predictor may of course identify a different service.

Service Agent : Each service agent is an extended version of the standard information agent platform developed by the GraniteNights project [1]. Each agent implements a query interface based on RDF Query By Example (QbEX) [1]. When the Environment Agent predicts that a user will use a service in the near future it alerts the relevant service agent. Once the service agent receives such an alert it generates a recommendation message based on the user profile obtained from the User Agent.

User Agent : The user agent manages interactions with the user through the client application; in addition, it handles the user

Context	Rules	Sup.	Conf.
Environment Rules	[restaurants]->[cinemas] [seminars]->[pubs]	41% 58%	100% 100%
New User	[restaurants]->[pubs]	2%	100%
Returning User	[restaurants]->[pubs]	**5%**	100%
Discovering a New Service	[restaurants]->[pubs] [restaurants]->[evenings]	3% 1%	70% 20%

Class	Proximity Rules
cinemas	location in {home} ˆdayOfWeek in {fri,sat,sun} ˆtime=17:25 (15)
pubs	location in {home,office} ˆdayOfWeek in {mon,thu,fri,sat,sun} ˆ15:45<=time<=17:15 (36)

Figure 3. Example Sequence & Proximity Rules.

profile. The queries contained in the profile (encoded as RDF) represent previous examples of user behaviour and include appropriate ontology references.

3 Discussion

This paper has presented an intelligent multi-agent system for context-aware service delivery and recommendations to mobile users. A system with very similar goals is ACCESS [2] in which context is considered an aggregation of the user's location, their previous activities and their preferences. RECO extends the ACCESS model of context by attempting to situate each atomic service task within a sequence, and by learning a user's preferences over time. By using environment rules RECO is still able to provide service recommendations even when a user has just joined the system.

Another similar system is ClixSmart Navigator [7] which implements a Navigation Server between a content store and a WAP gateway. In RECO we replace the standard browser with an agent-based client which allows the dispatch of recommendation messages without an explicit request from the user. Moreover the client is integrated into the device and is totally transparent to the user (see Figure 1).

The work described in this paper has successfully employed adaptive techniques to recommend services to a user based on past service behaviour and known preferences. The use of sequence analysis techniques enables the system to predict which service a user will access in the near future, while proximity rules allow the system to deliver personalised information if the user accesses a service at the right location and time. By capturing two different models of user behaviour (environment sequence rules and user sequence rules) the system can deliver service recommendations even if a user has never accessed the system. Our work to date leaves room for future expansion. Recommendations from newly visited services are not personalised, as there is no existing user profile. One strategy would be to compare a new user's developing sequence rule model with those of other users who also have more developed user preference profiles. Using techniques similar to those used in collaborative recommendation systems [6], aspects of those preference profiles could then be shared with the new user.

REFERENCES

[1] G. AA. Grimnes, S. Chalmers, P. Edwards, and A. Preece, 'Granite nights - a multi-agent visit scheduler utilising semantic web technology', in *Seventh International Workshop on Cooperative Information Agents*, (2003).

[2] C. Muldoon, G.M.P. OHare, D. Phelan, R. Strahan, and R.W. Collier, 'Access: An agent architecture for ubiquitous service delivery', in *Proceedings Seventh International Workshop on Cooperative Information Agents (CIA)*, (2003).

[3] E. Pignotti, *Intelligent Adaptive Service Delivery*, MSc dissertation, University of Aberdeen, 2003.

[4] P. Resnick and H.R. Varian, 'Special issue on recommender systems', *Communications of the ACM*, **40**, (1997).

[5] S. Salzberg, *A nearest hyperrectangle learning method.*, volume 6, 277–309, 1991.

[6] U. Shardanand and P. Maes, 'Social information filtering: Algorithms for automating 'word of mouth'', in *Proceedings of the ACM Conference on Human Factors in Computing Science*, (1995).

[7] Barry Smyth and Paul Cotter, 'Personalized adaptive navigation for mobile portals', *ECAI 2002*, (2002).

[8] Ramakrishnanan Srikant and Rekesh Agrawal, 'Mining sequential patterns: Generalizations and performance improvements', Technical report, IBM Almaden Research Centre, (1999).