UNOBTRUSIVE ACQUSITION OF USER INFORMATION FOR E – COMMERCE APPLICATIONS

Oshadi Alahakoon, Seng Loke, Arkady Zaslavsky
School of Computer Science and Software Engineering, Monash University,900 Dandenong Rd, Caulfield,3145, Australia

Keywords: Unobtrusive Interfaces, user modelling, e-commerce

Abstract: E-commerce has become a common activity among many people. Although widely used, the interfaces which users communicate with e-commerce systems are still at an early stage of development in terms of intelligence and user-friendliness. Unobtrusiveness is recognized as one of the most important desired attributes of an intelligent and friendly interface. In this paper we describe our work on an information architecture to minimize obtrusiveness. A layered information architecture supported by a structured user profile model is described in the paper. As example scenario is presented to clarify the new architecture and the development of a cost model for measuring the level of obtrusiveness is discussed.

1 INTRODUCTION

Recommender systems are being used by increasing number of commercial websites to guide consumers locating the products they will like. To provide such personalized results, recommender systems require information about user preferences and needs. Although the obvious method is asking from the user, this will result in user filling out lengthy forms or conducting long dialogs with the system. Such time consuming activities may make the system unpopular with the user. Therefore these interfaces providing personalized services need to be unobtrusive. Such unobtrusiveness may be achieved by minimizing direct user inquiry and providing users with an easy to use interface.

Although the most reliable first hand information is obtained from direct user inputs, it compromises the unobtrusiveness. As a solution we can use other less reliable information sources as web contents, stereotypes or prediction techniques based upon available information. Therefore it is apparent that with unobtrusiveness the precision of the user information is reduced. In our model we attempt to achieve a balance between unobtrusiveness and information precision.

eHermes is a web based multi-agent system (Jayaputera et al., 2002, Alahakoon et al., 2003), which is currently being developed at Monash University. It has a flexible and extensible open architecture, which can adapt to changing environments. eHermes helps users with their information needs such as financial services and online shopping for goods and services.

Our work is related to the on going work of the personalization component (front-end) of the eHermes system. The main challenges of this component are of two fold.

1. Unobtrusive acquisition of user information
2. Building and maintenance of adaptive user models

The information (described in 1) is required for construction of long term, reusable user profiles and also to identify current user need within a given application domain. Then storing and management of such acquired information as user profiles is required. Such profiles are organized (structurally) to be used in recommending online goods and services in different domains. (eg. Real estate, Insurance polices and Purchasing various commodities). In that profile structure, some components are reusable irrespective of the application domain.

In this paper we focus on the unobtrusive acquisition of information from the users. We identify 'levels of obtrusiveness' in information acquisition from a user, and relate these levels to a user profile with a layered architecture. Therefore the layers of the user profile represent different levels (or degrees) of obtrusiveness that will be 'forced' on a user during information acquisition. Using this definition, we present a model to

Alahakoon O., Loke S. and Zaslavsky A. (2004).
UNOBTRUSIVE ACQUSITION OF USER INFORMATION FOR E–COMMERCE APPLICATIONS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 3-8
DOI: 10.5220/0002633400030008

minimize the obtrusiveness by acquiring certain information from other sources and maintaining user information in structured profiles.

We structure our user profiles to hold user information belonging to different application domains in separate modules. Application domain independent information is to be stored in a common module. For example an individual using eHermes to seek for a recipe need not to re-enter some information (eg. highly obtrusive user demographic data), when he/she needs a recommendation for a car insurance policy. In that context, our system contributes to unobtrusiveness by not collecting the same information again.To demonstrate our model, we use a recipe recommender as an example scenario.

The rest of the paper is organized as follows. Section 2 describes past work, which are related to our research. Section 3 introduces our information architecture and the layered user profile model. Section 4 describes the user interface, which makes use of the new architecture described in section 3. A cost model for measuring the level of obtrusiveness is also discussed in section 4. Section 5 provides the concluding remarks with a discussion of the future work.

2 RELATED WORK

Personalized user adaptive systems are used in many application areas, as information filtering and retrieval, email filtering, recommendations in e-commerce and intelligent user interfaces. These systems obtain user preferences through interaction with the users, build user models and utilize these models to provide users with customized results. In long term they learn about the individual user and adapt themselves to give more personalised results.

The work we describe here lines up with the e-commerce recommendation systems. In addition to precise recommendations, our intension is to provide users with an unobtrusive interface, which makes system-user interaction an enjoyable one. Recent work on recommender systems includes personalized systems recommending music (Shardanand and Maes, 1995), electronic TV programme guides (Ardissono et al., 2001), restaurant recommendations (Tewari et al., 2000, Burke, 1999, Thompson et al., 2002), information retrieval (Middleton et al., 2001, Balabanovic and Shoham, 1995, Marko Balabanovic and Shoham, 1995), real estate (Shearin and Lieberman, 2001) and many other application areas.

The user profiles created in above systems are only to be used in particular applications. In that context, users have to employ different systems for their different information needs. Users have to disclose their information to each of these applications. Again these users need to get familiar with various user interfaces. To avoid such efforts, we propose a single system, with the ability to create and maintain an adaptive and application independent user profile. The profiles created in our model hold some common information for number of application domains.

In addition to above application dependent user modelling systems, there are user modelling shell systems (Orwant, 1991, Kobsa and Pohl, 1995, Kay, 1995). Shell systems maintain knowledge about users and assist interactive software systems in adapting to their current users by providing assumptions about user requirements.

Generally shell systems too maintain different user profiles for different applications. DOPPELGANGER (Orwant, 1991) is a server based shell system which has centralised user profiles for providing applications with assumptions about user behaviour. User information acquisition in DOPPELGANGER is done from various resources using number of techniques. Some of this information is gathered using sensors. This information is application independent. As most of the user modeling data that is useful for an application, remains application specific, this may not acquire useful information (Pohl, 1996). In our model, although there is a single user profile for all the domains, it is a collection of application specific user profiles.

The uniqueness in our model is in the use of 3-layered information architecture, which maintains user information according to obtrusive levels. Such information structuring gains more control over the user information. Using the cost model (section 4) together with structured profiles, it is possible to control the obtrusiveness of a particular user session. Again such structured profiles could be used to impose different security levels over user information. For example user information in the first level (more personal and private) of the profile structure can be regarded as confidential. Access of such information could be controlled using different authority levels.

3 INFORMATION ARCHITECTURE

In this section we discuss the categorization of information in to different levels and the structured user profile architecture.

3.1 Categorization of User Information

As our model is used in many domains for different user needs (using possibly different user interfaces) the profile generated should have both general and domain dependent information. We categorized the information requirement into 3 separate levels.

1. User Classification information – domain/ request independent(Eg. Demographics).
2. Domain dependent information – long-term user needs in the problem domain.
3. Request dependent temporary information – spontaneous/temporary user needs

In our proposed information acquisition model, all required information is categorized into the above 3 levels. Level 1 contains more personal & private information. The 2nd level of information is less private as it only describes the user with respect to a particular information domain. Level 3 is very general temporary information. We propose that a user would consider an attempt to acquire information at level 1 (classification information) as highly obtrusive while level 2 (domain) and 3 (request dependant) information as of lesser obtrusive nature. Therefore we allocate the degrees of obtrusiveness as high, medium and low to the information category levels 1, 2 and 3 respectively.

Since level 1 and 2 information can apply to a number of transactions we relate this information to a user's profile. This information represents a particular user or user group. On the other hand, level 3 information (although may be influenced by level 1 & 2) are only directly related to a particular transaction or event. To include the concept of this layered information architecture, we have developed a two-layered profile architecture of users (described below).

3.2 Structured User Profiles

As introduced above we create profiles that may be used in multiple application areas. Therefore we organize the structure of user profiles in to a two layered architecture.

The information required for these long term adaptive profiles are mainly of two types.

1. Information regarding general characteristics of the user – do not change values depending on the application area. Example demographic information.

2. User's likes and dislikes towards item attributes he/she is expecting to purchase-application dependent.

We call former attributes as Classification details and latter as Prediction details. Classification details are kept as domain independent. They are obtained from user inputs, past behaviour and using stereotypes. Some of the attributes never change as name, date-of-birth or gender; these are called static attributes. Attributes that can change over time such as address, occupation or income are called default attributes.

As an existing user make requests, for items from different application domains, system updates the default user classifications (if changed) and re-generates appropriate prediction details.

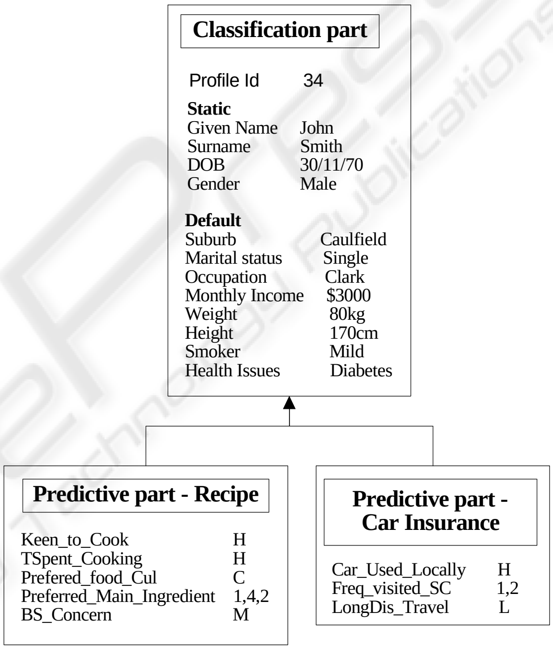

Figure 1: Layered user profile architecture

Each time a new request is received from a new application domain the profile grows by adding a new prediction layer instance. As classification details are domain independent, they are stored in the main module to be used commonly within different application domains.

Figure1 shows how two layers of user information in two different application domains (cooking recipe recommendation and car insurance policy recommendation) are connected to the main module. There is a main module consists of application domain independent user information. Rests of the modules are for different application domains holding user information corresponding to that particular domain.

4 A USER INTERFACE FOR OPTIMISING UNOBTRUSIVENESS

In user modelling literature, unobtrusive information (about the user) gathering is considered as one of the main challenges. In our work we take several measures to minimize system intrusiveness.

1. Ask for user inputs only when unable to use an alternative information source.
2. Ask clear and short questions that are quickly readable.
3. Provide options/possible answers, so that the user can reply with just a mouse click.
4. Finally, include user selected and system selected preferences with the results.

To obtain optimal unobtrusiveness, the following indirect sources are used in our system.

1. A well organized domain database
2. Stereotypes
3. Related information repositories - Eg. online recipes, food nutrition information, supermarket websites
4. Past user behaviour – Accepting or rejecting a system recommendation

4.1 The Unobtrusive user interface

The goal of our user interface is to acquire as much information as possible regarding the user's requirement(s) and preferences. The challenge is to acquire this information whilst minimising obtrusiveness.

The system maintains a static question graph with all the questions it needs in order to determine the most suitable recipe for a particular user. Questions in the graph may belong to any of three information layers described in section 3.1. The answers to the questions are used to filter out the large number of recipes. The most suitable once are presented to the user, with the option of browsing through a list of close matches. The question graph consists of

(a) A standard set of questions representing the domain of interest

(b) A set of links providing directions over possible alternative question sequences.

When a user logs into the system the questions will be activated. Depending on the level of obtrusiveness (calculated using a cost model - section 4.2), system decides either direct acquisition (i.e. ask the user) or indirect acquisition of answers (using one or more of the sources listed earlier). Progression of questioning is described with an example in section 4.3.

4.2 The cost model

A cost model has been developed to measure the unobtrusiveness during a user interaction session. We are hoping to use this as an optimizer to provide system users with unobtrusive interactions. The cost model is described below.

Each question is allocated a level of obtrusiveness (L), according to the information categorization level. This value is 'high' for level 1, 'medium' for level 2 and 'low' for level 3. Each question can obtain its answers from different resources, namely user input (UI), stereotypes (St), past user behavior (PUB), Information Repositories (IR) and available profile information (PI). Depending on the information source used, the degree of obtrusiveness (Obs), the certainty of values obtained (UN), and the time needed for acquisition (T) will vary. We calculate the cost incurred for each information acquisition act, by identifying values for these three parameters. The information source dependant values for the parameters are given in table 1.

Since the cost of question q_i will depend on the three values and the level of the question (within the profile),

$Cost(q_i) = L\,(a_1 \times Obs + a_2 \times UN + a_3 \times T)$

where $a_i \in N$, $i \in \{1,2,3\}$, $0 \leq a_1, a_2, a_3 \leq 1$ and $a_1 + a_2 + a_3 = 1$; L is level of obtrusiveness with in the profile.

Table 1

Info-Source	Obtrusiveness (Obs)	Uncertainty (UN)	Time (T)
UI	High	Low	Low
St	Low	High	Medium
PUB	Low	Medium	High
PI	Low	Medium	Low
IR	Low	Low	High

Costs Obs, UN and T can take values low, medium or high. For example values 0.3, 0.6 and 0.9 can be used for low, medium and high respectively. The total cost of a query would be,

$$\sum_{1}^{N} \text{Cost}(qi)$$

where N is the number of questions required to satisfy the query.

Values in table 1 demonstrate that we can build reliable profiles within a short time span by being highly obtrusive (or vice versa). The cost model can be used to adapt to the context (whether priority should be unobtrusiveness or reliability and/or speed). ie. using a maximum allowable total cost (as a threshold) the system can select the information sources for satisfying the questions, maintaining the cost within the threshold value.

4.3 An example scenario

In our example scenario, system interacts with the user in an unobtrusive manner and recommends recipes from a large recipe collection. We developed a database consisting of 50 recipes for testing purposes. These recipes are obtained from the AllRecipes (AllRecipes) website. The recipes are described in terms of user needs. Some of the attributes are Dish Type (eg. Main, Dessert, Appetizer etc), Meal Type (Lunch, Dinner, etc), Tradition (eg. Chinese, Indian, Malaysian, etc), Food Type (e.g. Beef, Fish, Poultry, etc).

4.3.1User Interface

In our implementation, the questions are displayed one after the other in dialog boxes. The user can either answer or ignore the questions. Most of the questions come with options (possible answers), for the user to select with a mouse click. A subset of questions and answers in a session is shown in the figure2.

4.3.2 System Questions

The questions ask may belong to different 'levels' of information categories as described in section 3. For example questions and their possible information level (given within brackets) are as follows.

Q1: What meal Time? (3)
Q2: What is the occasion? (3)
Q3: What season? (3)
Q4: What dish? (3)
Q5: What traditional food preferred? (2, 3)
Q6: Any health issues? (1)
Q7: What cooking times preferred? (2)
Q8: Any preferred main ingredient? (3, 2)

It is possible for certain questions (Q5 and Q8) to belong to two or more levels. In such a situation, the Cost calculation will require a pre-defined fraction of membership to each level.

Figure 2 demonstrates the question graph with two alternative routes. The path may divert to another node depending on the answer/information for a particular question. For example (figure 2) for different answers to Q1 (meal time) the path will be directed to another node as some questions (eg: Q4) becomes irrelevant.

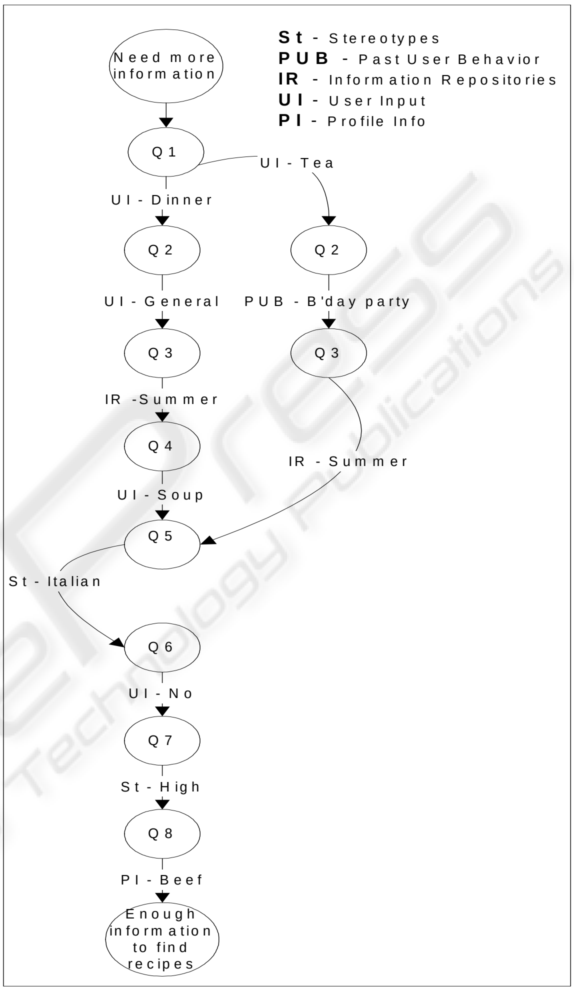

Figure 2: The question graph

5 CONCLUSIONS AND FUTURE WORK

The paper described some on going work on building an unobtrusive interface to an e-commerce system. The main contributions of this paper is the concept of identifying levels (or layered) information architecture and the categorization of the user requirements into this architecture. The structured user profiles (Alahakoon et al., 2003) described, supports this model and thus provides the foundation for our cost model. The structured user profile provides the advantage of separating the domain independent information, thereby making it possible to use the top layer of the user profile for many domains.

ACKNOWLEDGEMENTS

Support for this project from the Australian Research Council (ARC) Linkage Grant LPO211384 and Microsoft is thankfully acknowledged.

REFERENCES

Alahakoon, O., Loke, S. and Zaslavsky, A., 2003. Modular and hierarchical user profile instantiation for e-commerce applications using mobile software agents. *The 14th Australasian Conference on Information Systems (ACIS).*

AllRecipes, http://allrecipes.com/.

Ardissono, L., Portis, F., Torasso, P., Bellifemine, F., Chiarotto, A. and Difino, A., 2001. Architecture of a system for the generation of personalized Electronic Program Guides. *Proc. UM2001 Workshop on Personalization in Future TV (TV'01).*

Balabanovic, M. and Shoham, Y., 1995. Learning Information Retrieval Agents: Experiments with Automated Web Browsing. *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous,Distributed Resources. Stanford,CA.*

Burke, R., 1999. The Wasabi Personal Shopper: A Case_Based Recommender System. *In Proceedings of the 11th National Conference on Innovative Applications of Artificial Intelligence.* 844-849, AAAI,.

Jayaputera, G., Alahakoon, O., Cruz, L., Loke, S. and Zaslavsky, A., 2002. Assembling Agents On-Demand for Pervasive Wireless Services. *Proceedings of the Second International Workshop on Wireless Information Systems.*

Kay, J., 1995. The UM toolkit for reusable, long term user models. *User Modeling and UserAdapted Interaction.* 4(3), 149--196.

Kobsa, A. and Pohl, W., 1995. The user modeling shell system bgp-ms. *User Modeling and User-Adapted Interaction.* 4(2), 59--106.

Marko Balabanovic and Shoham, Y., 1995. Learning Information Retrieval Agents: Experiments with Automated Web Browsing. *Proceedings of the AAAI Spring Symposium on Information Gathering from Heterogeneous,Distributed Resources. Stanford,CA.*

Middleton, S. E., Roure, D. C. D. and Shadbolt, N. R., 2001. Capturing Knowledge of User Preferences: Ontologies on recommender systems. *In Proceedings of the First International Conference on Knowledge Capture (K-CAP2001), Victoria, B.C. Canada.*

Orwant, J., 1991. M.S. Thesis, Doppelgnger goes to School: Machine Learning for User Modeling. *Program in Media Arts and Sciences.* MIT.

Pohl, W., 1996. Learning about the user -- user modeling and machine learning. *Proc. ICML'96 Workshop Machine Learning meets human-Computer Interaction.* 29--40.

Shardanand, U. and Maes, P., 1995. Social Information Filtering: Algorithms for Automating ``Word of Mouth". *Proceedings of (ACM) (HI) 95 Conference on Human Factors in Computing Systems.* 1, 210--217.

Shearin, S. and Lieberman, H., 2001. Intelligent Profiling by Example. *Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001) ,Santa Fe, NM,.* 145-152.

Tewari, G., Youll, J. and Maes, P., 2000. Personalized Location-Based Brokering Using an Agent-Based Intermediary Architecture. *In Proc. of the Int. Conf. on E-Commerce, Seoul, Korea,.*

Thompson, C. A., Göker, M. and Langley, P., 2002. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research.*