Abstract - User profiling has been used to provide personalized e-services. In our previous work we have highlighted the advantages of a modular approach to developing user profiles. This approach separates the information to be stored into levels according to the dependency level on the customer or transaction. Therefore the customer’s personal information is stored in a separate layer while individual transaction information is stored in another layer. A middle layer stores information generalized from the individual transactions, which are used to reduce the need for intrusive information extraction from a customer during a subsequent transaction. In this paper, we highlight the advantages of the modular user profiles when used in a de-centralized setting. The separate layers provide a flexible structure, which provide the possibility of locating in vendor, server or client sites. A decentralized profile architecture is presented in the paper, and the advantages are discussed using possible scenarios.

I. INTRODUCTION

In today’s e-commerce applications, providing personalized services for consumers has become a fundamental and essential feature. Personalization makes consumer feel that they have a unique personal relationship with the system, which is anticipated to increase the number of website visitors and also provide more satisfaction thus helping to retain a higher percentage of existing customers. Since more domain dependent user information helps to make better recommendations, many web sites currently provide personalized services in narrow e-commerce domains; for example personalized web stores [3], restaurant recommendation [4], personalized real estate finder [17], personalized TV guide generation [5] and many others. Although the techniques used in these systems are suitable to be used in other domains, the information used in the profiles are not reusable in any other domains. Therefore each time the user changes the domain he/she has to register in a new web site. There are many instances that these users have to provide the same information repeatedly to different websites. A solution to this problem is the development of user profiles which can support multiple domains. To be practical and adaptable to the commercial environment, these models must have the ability to function in a decentralized setting.

Applications such as Microsoft passport [12], Liberty alliance [11] can also be considered as alternative solutions. Although usable in decentralized applications, these are simple data stores without inferencing ability and do not satisfy the requirements of personalization. Information gathered to describe user requirements in a particular application (domain) can in many instances also be used to infer knowledge about other domains. Although the user modeling shell systems [9] [7] [13] are made to serve in more than one application [8] they do not have the ability to use decentralized information. Instead of maintaining distributed data regarding individuals, many researchers [6] [13] [16] suggest using single electronic user profiles that can serve multiple applications. Based on such work, we propose a novel method of generating user profiles, which have components that can be used in multiple domains or applications in a decentralized environment.

The rest of the paper is organized as follows. Section II describes the modular architecture in detail. Section III describes expansion of proposed model in to a decentralized environment. Section IV discusses the advantages of the model in a decentralized environment. Finally the conclusions and the future work is presented in section V.

II. PROPOSED MODULAR ARCHITECTURE

The main goal of profiling in commercial activity is to provide a better personalized service. Personalization requires the identification and the understanding of user needs and behaviors. We suggest that the user needs and behavior could be summarized in the following manner.
Although the level of personalization increases at the lower level (as shown in Figure 1) the value of such personalization can differ. More specifically, it may be preferable to be able to function on individual user behavior when focusing on the individual user. But when focusing on products, individual transactions may not be sufficiently generalized to be useful for predicting future behavior. As mentioned in the introduction, a user profile which can cater for the needs of multiple domains will provide many advantages. A profile built by combining the above two aspects (user focus and product focus) will have the ability of representing an individual customer’s needs/behavior in different domains. In the proposed architecture this combination has been carried out by linking A and C (in Figure 1) using a structured set of product features called a product feature hierarchy. The rest of section II describes a new architecture implementing the above discussed concepts. The proposed architecture is composed of two main components; product domain hierarchy and layered user profile. These components are described below.

A. Domain hierarchies

Personalization can be considered as the mapping of product features to user needs such that user needs are satisfied in an individual manner as possible. In the proposed model we list a set of features for each product in a domain. For example, the restaurant data set (used in entrée system) has 256 features which describe restaurants. Each application area is called a domain. Each domain may have super domains and sub domains. (Both restaurants and recipes domains are sub domains of food domain). Each item/product in a domain (say a particular restaurant R in the restaurant domain) is described using features (location, food type etc). Features can take binary values depending on the presence or absence within a given item. Product features are possible to be grouped in to classes, where more than one feature is describing an attribute of a product.

The main function of the domain hierarchy is to provide a mechanism for separating and representing an individual’s domain specific interests in the user profile. This is made possible in the proposed model by a module in the user profile representing the product feature hierarchy. An example of a product feature hierarchy is shown in figure 2.

B. Layered user profile

The second component of the modular architecture is the layered user profile. This profile is built by combining the user and products aspects described above (Figure 1). In order to combine, three types of information were identified from the user and products focus model (Figure 1). (i) Individual user behavior is independent from any product features or domains and therefore named as domain independent information. (ii) User group information combined with product feature classes results in providing domain specific ‘behavior’ for specific user groups; and are called domain dependent information. (iii) Finally individual transaction related information is called transaction dependent information. Figure 3 shows the 3 information types positioned in the layered profile.

Figure 2: Layered user profile

Category (i) information identifies an individual customer by personal characteristics. This information is mainly gathered during initial registration. Customer’s socio-demographic information (e.g., age, suburb, marital status, education etc) is gathered under this category. Category (i) information layer is common to all domains and applications.

Category (ii) information represents an individual customer’s behavior in a particular domain. As discussed above, the domain characteristics are represented by the corresponding feature values from the domain hierarchy. The justification for this layer is the argument that an individual has behavioral patterns which are domain dependent. There can be related or unrelated domains with shared features and category (ii) facilitates the representation of common behavior using such shared features. Initially the individual characteristics of the customer (layer 1) are used to filter out relevant information for layer 2 from a classified set of socio-demographic information.

A similar method is used in Lifestyle Finder [10]. Lifestyle finder maps the incoming user to an existing socio-economic group (cluster of users) for marketing purposes. Then complete the user profile by existing cluster attributes. The initial identity information obtained at the registration can be used to map the user to an existing cluster. Then the cluster information is filled in to the profile under relevant domains and sub-domains. The proposed method is similar to the use of the PRIZM [14] database used in Lifestyle Finder. If the above process fails to provide a value for any layer 2 feature, a default value of ‘NULL’ is used. These initial layer 2 values are updated using generalized (say averaged) transaction values (described below). As such layer 2 will adapt to the individual behavior of a customer, for the particular domain. Category (iii) consists of individual transactions records. The purpose of this category is to capture individual behavior at each transaction level such that these can then be generalized to be upgraded to layer 2 domain behavior.

Features form feature classes, and feature classes describe the user behavior in each domain. If the feature class is “Preferred food tradition”, where each feature, (say Italian, Indian, English, Chinese etc.) is assigned a binary value 0 or 1 depending on whether the user prefer it or not. Then the domain description “User’s preferred food tradition” in level 2 of the profile has few user preferred food traditions (say, Chinese, Korean and Burmese). After each user transaction, all the positive features are averaged and filled in to the level 2 of the profile. Therefore the level 2 information changes from initial general cluster information to more specific personalized behavior of a particular customer. Figure 4 illustrate the generalization of layer 3 data to layer 2. Here domain $D_j$ has the feature value $F_j = \mu_{F_j}(v_{jk})$ where $i$ is the domain number, $j$ is the feature number and $k$ is the transaction number. $\mu_{F_j}$ is the generalization function for the feature $F_j$.

Figure 3: Updating level 2 data using transaction data

III EXPANSION AS A DECENTRALIZED MODAL

Since the layered architecture enables the maintenance of profile as components, it has the ability to easily be implemented in a decentralized environment. Such decentralization enables profiling users across multiple domains.

A. Multiple domain or cross domain use of profiles

As explained earlier, a single domain refers to a single application area. Therefore multiple domains refer to related or unrelated collection of different domains. These domains could exist within different products/services based departments in the same organization (e.g., Apparel, home ware, etc). It is also possible for multiple organizations providing different products and services to agree upon sharing components and information in the user profiles. The modular nature of the profiles is structured to facilitate such functionality.

B. Expanding the profile to support multiple domains

In a decentralized and/or multi domain application, although the information in the profile is the same, the profile has to cater for multiple domains. Since in the
level 1 contains socio demographic information, independent of user transactions it could be shared by different domains. This could be implemented with authorization from the customer, when registering for a second domain. Such information sharing avoids user effort for re-registering to a new system and hence improves the unobtrusive behavior of the system. Integration of level 2 and the subsequent processing will be similar to the descriptions in section II. The main difference would be the use of common features across domains. When such features exist, these could be shared across domains reducing the need for requesting from the user.

IV. ADVANTAGES OF THE DECENTRALIZED MODAL

In this section we will discuss the benefits from the proposed model, especially in a decentralized application. The following subsections address different problems and questions that arise in decentralized applications. Using recipe and restaurant (from Entrée system) datasets, we demonstrate how the proposed model could be considered as a solution to these situations.

A. Decentralized approach in the proposed model

We attempt to describe the decentralized model from the customer’s perspective, considering that the goods and services offered by different applications have to be finally satisfy the customer. Therefore the proposed model identifies the following characteristics in a decentralized application/model.

Remark 1. In a decentralized model, many different products and services need to be offered to and handled/managed by the customer. All these products and services can be grouped, categorized and ‘linked to each other’ according to the needs of customer. For example all products and services could be linked in a hierarchy with a root “items and services”. The first level of nodes can refer to a customer’s basic needs such as food, clothing, entertainment etc. Initial structure is created by looking at the customers and their needs, without considering the applications involved. The applications can then be considered as a second stage and linked to the appropriate node/branch in the hierarchy. One of the advantages in this is that any new application (product/service) could easily be accommodated within the existing structure.

Remark 2. A customer is considered as an individual with a separate identity and certain specific individual needs. The ideal user model should be able to identify each and every customer separately and cater to the individual needs. Although there is a long way to go before achieving this, the level 1 in the proposed model cater for the need of individuality.

Remark 3. Although customers are individuals with specific and different needs there are certain commonalities in behavior according to application domains. Level 2 of the proposed profile is an attempt to represent this behavior. The individuality of the customer is maintained by linking to level 1 while the application domain specific behavior is captured by the domain hierarchy. Therefore the applicability to a decentralized situation is facilitated by the level 2.

Remark 4. Although level 2 results in the profile being able to cater for different domains decentralization is achieved with level 3. Level 3 is really a structured store of transactions for an individual customer, in a particular domain. Since the different applications may run in different environments and varying systems, it becomes necessary to structure the transaction information to a common format. The common structure is achieved through the domain hierarchy.

As described in section II, B the transactions are averaged or generalized periodically and used to update level 2. Therefore the decentralized transactions are used to fine tune the domain dependent characteristics of the user, thus capturing and passing on the users varying behaviors in the decentralized applications in to the level 2 of the profile. Since level 2 and 1 are used in supporting applications which require the user profile, this can be considered as a user profile feeding from, and supporting decentralized applications.

As described earlier, level 2 information is further restructured according to common features in the product hierarchy. As such, if and when common features are provided with values from one domain, this can benefit the customer when transacting with the other domain. Therefore the proposed profile has the capability of enrichment of profile data from one domain (or site), which can provide cross domain benefits.

B. Analysis of the proposed model as a solution to common problems in decentralized user models

In section IV, A we have described the ‘decentralized’ nature of the proposed model and demonstrated several important aspects using examples. In this section we describe the potential of the model for addressing a number of problems due to decentralization.

Making use of information captured in one context (domain) to provide value in another domain.

The domain hierarchy (or the product feature hierarchy) plays a central role in providing a common ground for information sharing by the possible diverse client applications. The basis of information sharing will be the ‘common features’ identified in the product hierarchy (Figure 5). The product hierarchy is used as a structure.
for information representation in layer 2, thus passing on the information sharing capability to the profile. Initially the level 2 values will be filled up from domain specific stereotypes as described in section II. The real value of decentralized information sharing will be seen once the user starts interacting with the system. The user specific peculiarities in behavior will thus be captured in one context (or domain) and be available for other contexts which have common features. An important advantage of this is that if the customer does not show stable behavior across related domains the model will capture this in the common features. Such information will contradict the stereotypic ‘opinions’ initially formed thus improving the level of personalization. There is more work required to identify ways of managing conflicting situations in common features.

Figure 4: Common features from two domains

Adapting the user model to new user behavior

The regular updating of level 2 from level 3 information results in the user model automatically adapting to the users current behavior patterns. Due to the generalization mechanism in level 3, the effect of a spontaneous and out of character transaction are diluted before passing on to level 2 as the individual’s ‘general’ behavior in the domain. If level 3 is maintained as a moving window of the latest ‘X’ number of transactions, the results will be an adaptation of the generalized profile to the customers current behavioral trends and a fading away or forgetting trends (if behavior changes over time).

Resolving conflicts and trust among different participating client applications

Conflicts may occur especially when client sites from different organizations participate in a decentralized profiling system. In the proposed system the main mechanism for reducing such conflicts is the initial agreement by the client sites to pass and share the information via the product hierarchy. The client site will ‘see’ the branch of the tree to which they belong, up to the root, thus will be aware of their position in the tree. A major obstacle in such a system will be the transaction information, which will be confidential. Therefore the customer’s authority will be required. But it must be highlighted that the organization will be providing transaction information (with customer approval) which is less confidential compared to the customer’s private information. Personal information (level 1) will be provided to the system only by the customer, if and when agreeing to participate in the system in anticipation of a more personalized service.

It is anticipated that by restricting the inter client communication to information passing via a structured product feature set most conflicts will be eliminated. But it is expected that such a system will be more successful within a single organization context due to the issues that will arise in collaboration and information sharing. But it must be also highlighted that such obstacles are not specific to the proposed model, but general to any inter organizational information sharing collaborations.

Privacy Issues

The layers in the propose profile are organized according to domain dependent nature of the of information content. We have interpreted the domain dependent nature as being positively related to the level of personal nature (or confidentiality) of the information. Due to the modular profiles, it is possible to maintain the highly personal information in level 1 in a single location in a secure server. Level 3 information can be maintained in the client site, thus overcoming the need of sharing actual transaction information. The information shared will be summarized versions of the feature values and as such will be of less privacy risk (eg: monetary information not included). We are also considering the possibility of maintaining the level 1 of the profile with the customer, and carrying out interactions using a mobile agent. The positives and negatives of such an implementation need to be further investigated. Overall, it is believed that the modular nature of the profile makes it better suited for positioning in suitable locations thus reducing privacy concerns.

V. CONCLUSIONS AND FUTURE WORK

A modular user profile architecture was described in this paper and the advantages of the modularity for decentralized processing was discussed. The main contribution of the paper is the presentation of the advantages of the proposed architecture, especially in a decentralized environment. A simple prototype of the model has been built and experiments with different datasets are being carried out. A recipe dataset and a restaurant dataset (from Entrée recommender system [4]) were used to demonstrate several issues in this paper. Work is currently in progress to evaluate the basic model by applying it to improve an entropy based method [15].
for information gathering from customers. We have also reported on the advantages of this architecture for improving unobtrusiveness [1] [2]. There are many issues still to be resolved regarding this new architecture. The rules for initializing layer 2 information need to be further investigated. Averaging or generalization functions for level 3 transaction information need to be identified for varying feature types. Our current work is focused on strengthening the layered architecture by testing with a number of datasets to highlight the practical advantages of the model.

REFERENCES


