USER MODELLING SERVER FOR MOBILE AND ELECTRONIC SHOPPING

Anastasios Savvopoulos and Maria Virvou
Department of Informatics, University of Piraeus, Karaoli & Dimitriou St, Piraeus, Greece

Keywords: User modelling server, e-Commerce, m-Commerce, Adaptivity, Clustering, Stereotypes.

Abstract: Recommending applications are used by many researchers as the main means of achieving personalization. However, the difficulty is to apply the same recommender and personalization techniques to a totally different system that belongs to a different domain and uses a different medium. Furthermore the difficulty level rises when we want to apply the previous recommendation techniques without changing major components or process attributes. In this paper we propose a server that can be used to achieve personalization to a product recommending system. The main advantage of this server is that works both for e-commerce and mobile commerce. We present a case study that we incorporated the user modelling server, which is a mobile e-shop, and discuss within the case the obstacles and breakthroughs we have achieved. The case study clearly shows that the same user modelling server can enhance e-shop application behaviour towards adaptivity and customer personalization.

1 INTRODUCTION

e-Commerce applications have become very popular since they provide easy access to all kinds of products. However, most of existing applications are generic and do not address specific needs, preferences and attributes of individual customers. A remedy to this problem can be achieved by web personalization techniques. As De Pessemier et.al. (Pessemier et.al. 2008) suggests, new technologies such as Internet, iDTV, and mobile applications create the possibility to advertise in a different, more attractive manner than the traditional commercial breaks. This leads to a rise of interactive commercials on websites, banners on the internet, and commercials on mobile devices. Thus the market moves towards more converged architectures corresponding to different types of mediums such as mobile phones, internet. A solution to convergence can be found within user modelling servers (Kobsa, 2001). In this paper we will propose a user modelling server for e-commerce and m-commerce applications that incorporates a clustering algorithm in order to create groups of similar users, concerning their needs and interests. The paper is consisted of five sections: Introduction, Related Work, User Modelling Server Architecture and the Case Study.

2 RELATED WORK

There are many e-shop applications that try to make recommendations using many techniques. These techniques usually involve the construction of user models that are either based on explicit user information or on data about the user behaviour that is collected implicitly by the system. User models can be constructed using many techniques like stereotypes (Rich 1979), adaptive hypermedia (Brusilovsky, 2001) or individual user models.

Another approach that goes beyond attributes exploitation like the work mentioned previously is done by J. Alspector et.al. (Alspector et.al. 1997). This approach builds a CART network with the help of three main adaptive techniques: feature-based, clique-based and linear model. Their study showed that an effective movie-recommendation system should combine all these approaches in order to maximize performance. A very interesting technique also, based on a rating system has been conducted by Q. Li and B. M. Kim (Li & Kim, 2004). The system acquires rates and then calculates fuzzy inferences and extracts similarities between users. Their method proved very successful according to the evaluation presented. Another recommending system that uses clustering to provide recommendations has been made by Ardissono and Torasso (Ardissono & Torasso, 2000). In their

Savvopoulos A. and Virvou M. (2009).
USER MODELLING SERVER FOR MOBILE AND ELECTRONIC SHOPPING.
In *Proceedings of the 4th International Conference on Software and Data Technologies*, pages 297-302
DOI: 10.5220/0002243202970302

research they created a web-store shell using dynamic user models.

In the field of mobile-shopping little work has been done. An interesting approach has been made by Billsus et al (Billsus et al 2002). In their work they have created and adaptive interface for mobile devices that makes personalised suggestions based on search questionnaires. Despite that fact that their work can be very efficient, it mainly focuses on filtered lists of suggestions on various areas such as restaurants, presented to the user's mobile device. Their adaptive user interface lacks the generality of a user modelling server that can be applied in both internet and mobile shopping applications.

Despite the fact that all the above systems provide users with recommendations, they are so domain and problem depended that lose the ability to be applied with easy in different media or products. In this way every time a new application is built a new architecture must be constructed from scratch in order to address the specific application's problems. Few systems have tried to create user modelling servers that can be applied to any product and help users in an adaptive way. A very important research on this field has been made by Brusilovsky et al. (2005). Their research is focused on user modeling and adaptation in distributed E-Learning systems. They describe CUMULATE, a generic student modeling server developed for a distributed E-Learning architecture, KnowledgeTree. They also introduce a specific, topic-based knowledge modelling approach which has been implemented as an inference agent in CUMULATE and used in QuizGuide, an adaptive system that helps students select the most relevant self-assessment quizzes. We also discuss their attempts to evaluate this multi-level student modeling. On the other hand our user modelling server applies to entirely different domain. Its role is not distribute student models but create adaptive customer models in real time as the customer navigates through the specific application.

Another significant work towards convergence has been done ns Personis by Kay et al. (Kay et al., 2002). Personis is a server for user models that every user can control his/her personal user model. Their user model server gives every application its monitoring a different view of the user model database. This structure of this database is based on objects. Our system has a user model server but instead of just control it creates user models dynamically. Our user modelling server provides components that help systems create recommendations and effectively classify new customers without any prior knowledge of their tastes.

In this paper we propose the main architecture of a server that can be applied in any product recommending application in order to transform it into an intelligent application. This framework is product and media independent. The main advantage is that it can be applied on any product recommending application without consideration of the products used such as personal computers, mobile phones or even cars. The architecture of the user modelling server includes user modelling techniques, intelligent user interfaces and clustering algorithms in order to produce recommendations. We also propose the basic steps on how to build the user modelling server and present two case studies in which we incorporated our server. The case studies sell movies. The first one is an e-shop applications and the second a mobile shop application. The incorporation of our user modelling server on both cases proves that this server can be easily applied on any product recommending system.

3 THE PROPOSED USER MODELLING SERVER ARCHITECTURE

The architecture of the user modelling server follows the logic of "two modules" (figure 1). The first module consists of components that perform reasoning about users' actions and users' preferences (indirect components). More specifically the first module consists of the following components:

Explicit Information User Profiles: information in a database about users provided to the system in an explicit way (answering interview questions or rating products). Every time a new user is registered in the application that incorporates the user modelling server, this component collects this information from demographic data, educational data and interest data that the user provides through the registration process

Monitoring Agent: includes information about user's interactions with the system. Monitoring in real time user actions throughout system usage. This component contains information about the main product categories that this user has visited, specific products that user visited, products that user moved in or out of his cart and products that user bought from the shopping cart. In this way this component, is product independent because it monitors user behaviour and not products characteristics. The Monitoring Agent component also contains a statistics database of all users' actions and features collected by the systems.

Clustering Algorithm Process: uses k-means clustering algorithm in order to group similar users. These groups are then processed and result in users' representatives that the user modelling server can use as the basis of users' interests. The k-means clustering algorithm takes as input the statistical data of all the explicit and implicit information that the system infers and collects about users. This data is saved on the statistics database of the Monitoring Agent. This data include the visits in products specific pages, products general categories, search queries made by the user, products moved to the shopping cart and products eventually bought. Every category and product characteristic is a calculated by the algorithm. However, this calculation is not based on degrees from products classification; instead the calculation is based on user behaviour while s/he interacts with the system. The data input for the k-means algorithm are calculated based on degrees from the features measured. An example general equation can be seen below. This equation measures interest degree in product category. Weights change, depending on the application that this framework is incorporated. However, general rules on weights apply on all applications that use this framework. For example, the weight concerning visits on a specific product category is always smaller then the corresponding weight of products moved to the cart. The first weight is smaller, because visiting a product may not mean that the user is necessarily interested in this product but that s/he is just browsing several products.

The equations above can be specialised to every product the application that incorporates the user modelling server. In this way user modelling server remains product independent. The k-means clustering algorithm processes these degrees and provides the system with groups based on similarity. From these groups representative feature vectors are extracted. The representatives' vectors work as group leaders and show the groups tendency to specific product features. These vectors are then compared with the vectors of products' characteristics and the closest vectors are extracted and saved to the Recommendations Database component. In this way the Recommendation Database component is updated dynamically as users navigate through the application.

$$DegreeOfInterestInCategory_a = \frac{VisitsInProductBelongToCategory}{VisitsInAllProducts} \quad (1)$$

$$DegreeOfInterestInCategory_b = \frac{ProductsPlacedInBasketBelongToCategory}{AllProductsPlacedBasket} \quad (2)$$

$$DegreeOfInterestInCategory_c = \frac{ProductsBoughtBelongToCategory}{AllBoughtProducts} \quad (3)$$

$$\begin{aligned} &DegreeOfInterestInCompany = \\ &W_a * DegreeOfInterestInCategory_a \\ &+ W_b * DegreeOfInterestInCategory_b \\ &+ W_C * DegreeOfInterestInCategory_c \end{aligned} \quad (4)$$

Double Stereotypes: dynamic double stereotypes from these representatives. Stereotypes, within the component, follow the rule "general to specific". These stereotypes form a hierarchy from top to bottom. This means a hierarchy is constructed with a low number of generic stereotypes at the top. As the hierarchy continues to bottom the stereotypes participating in the same level are constructed with more specific attributes. This hierarchy is continued until it reaches a certain point of complexity.

Server: The Server component is involved in the manipulation of all the information about a specific user, the stereotype that s/he belongs to and his/her more similar representatives. The Server also contains all users' models and manages them in order to give the right information to the components of the second module. In this way the server acts as a service of communication between indirect and direct components.

The second module consists of user interface components (direct components).

Incremental Initialization Process: acquires information from the server and provides the closest recommendations to new users or users that the system has little information about. Stereotypic information is used throughout this process, from the Double Stereotypes component, to fill the blanks of knowledge concerning the specific user. The main operation of this process is to choose whether a user, according to the actions made so farm, belongs to a group or not.

Recommendations: communicates with the Server component and provides the users with personalised

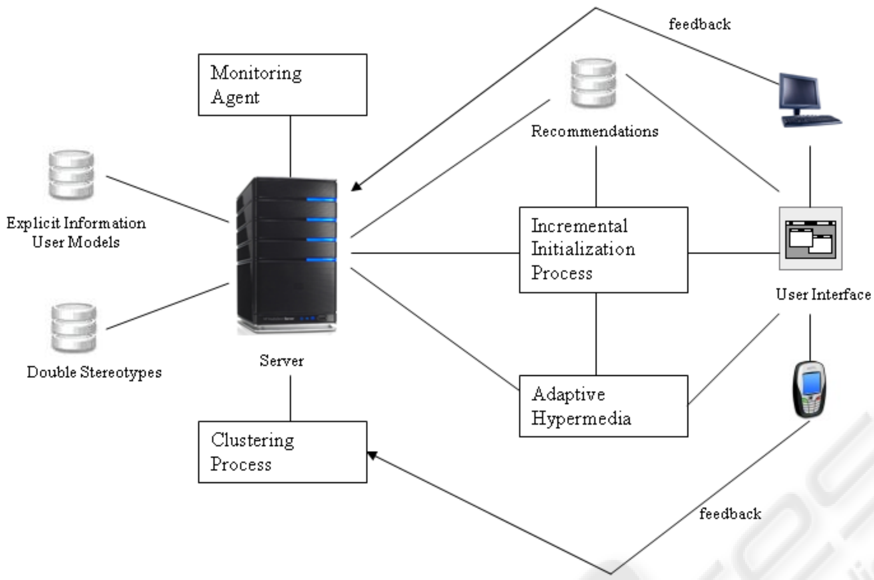

Figure 1: User Modelling Server General Architecture.

recommendations and advice. The Recommendation component is a database with system recommendations about products, mistakes or other personalised advice in a database. This main techniques used by this component are adaptive hypermedia and dynamic annotations. The Recommendation's database is built by the user modelling server adaptively in real time as the user provides the system with feedback along his/her interaction with the system.

User Interface: adjusts to the medium used automatically and changes according to the users' interests. The whole user interface appearance changes adaptively and personalises its behaviour according to the specific users' inferences and needs. The User Interface is an entirely separate component and adapts to any medium thus making the user modelling server medium independent. These two functions of the User Interface component create a unique personalised experience for every specific user, resulting in a friendlier and more efficient user interface. Except the Server this component communicates with the Recommendation and Incremental Initialization components in order to adapt the user interface accordingly.

Adaptive Hypermedia: component with product annotation ability and dynamic placing of recommended products. This component has the ability change the symbols of recommended products depending on the degree of the recommendation. For example, highly recommended products by the user modelling server are tagged with a different symbol by this component. On the other hand, lower recommendation degrees can result in different annotation product symbols. Application fonts and font-sizes can also be changed by this component in order to get user's attention and differ his/her behaviour. For example, if a user is greatly interest in a specific product feature then in adaptive hypermedia will enlarge the font of this feature in the product page, giving it the proper attention.

4 THE CASE STUDY MVISION

Our case study is a mobile shop that sells movies, called mVision. MVision is built on ASP.Net Mobile uses the mobile device resources of every customer to create an effective user interface for every customer. Figure 2 shows the main movies categories page. Next from the action-adventure movies category we have the implementation of the adaptive hypermedia technology with the symbol of . At the top left of the screen is the "cart" button that can be pressed to access the specific customer's cart. MVision is corresponding mobile shop of Vision.Com (Virvou et al., 2007).

For every customer mVision creates a different record at the database. There are two types of information saved for every user, the explicit and implicit. The explicit information is saved on the

Explicit Information User Profiles and the implicit are saved by the Monitoring Agent. In mVision every customer can visit a large number of movies by navigating through four movie categories. These four movie categories are: social, action, thriller and comedy movies. All customers have their own personal shopping cart. If a customer intends to buy a movie she/he must simply move the movie into her/his cart by pressing the specific button. After deciding which movies to buy a customer can easily purchase them by pressing the button "buy now!"

All navigational moves of a customer are recorded by the system in the statistics database by the Monitoring Agent. In this way mVision saves statistics considering the visits in the different categories of movies and movies individually. The same type of statistics was saved for every customer and every movie that was moved to the buyers' cart. The same task is conducted for the movies that are eventually bought by every customer. All of these statistical results are moderated from one to zero and saved in the statistics database. In particular, mVision interprets users' actions in a way that results in the calculation of users' interests in individual movies and movie categories. Each user's action contributes to the individual user model by implying degrees of interest into one or another movie category or individual movie. For example, the visit of a user into a movie shows interest of this user to the particular movie and its category. If the user puts this movie into the shopping cart this shows more interest in the particular movie and its category. If the user buys this movie then this shows even more interest whereas if the user takes it out of the shopping cart before payment then there is not any increase in the interest counter. Apart from movie categories that are already presented, other movie features that are taken into consideration by mVision are the following: price range, leading actor, director and price.

As a consequence, every customer's interest in one of the above features is recorded as a percentage of his/her visits in movie-pages similarly to Vision.Com. MVision like Vision.Com incorporates the AIN clustering algorithm. In mVision we also used the AIN representatives in order create double stereotypes that are saved in the Double Stereotypes component of our general architecture. The stereotypes concern both users and movies. Again, levels 1 are the more general stereotypes and level 5 the more complex ones. The more information the system has about a user the more complex stereotypes it uses, until it reaches the level 5. After level 5 the system uses an individual user model about this user. Double Stereotypes and individual user models use the Adaptive Hypermedia component to suggest movies (figure 3). Figure 3 shows the implementation of the adaptive hypermedia technology in the recommendations page. In this example this specific customer, according to his/her user model, likes these movies very much, so the user modelling server provides mVision with the right information and degrees of interest in order for mVision to be able to make these movies suggestions. The symbols of and show greater interest degree opposed to the symbol. MVision does not support an animated agent due to hardware limitations.

Figure 2: mVision User Interface from Movie Categories page.

Figure 3: Recommendations Page.

5 CONCLUSIONS

In this paper we proposed a user modelling server that can be used to achieve personalization to an e-commerce system. The main advantage of the user modelling server is that it can be incorporated as a component in different media such web or mobile systems. We presented a case study that we incorporated this user modelling server. The case study was on a mobile device e-shop application. This case study proves the independency of our user modelling server and how easily it can be incorporated to any kind of shopping system. Our case study also showed how the component can change the application's user interface in order to benefit from the adaptivity and suggest products more efficiently.

REFERENCES

Ardissono L, Torasso P., 2000. Dynamic User Modeling in a Web Store Shell. In Proceedings of the 14th Conference ECAI. Berlin, Germany, 621-625.

Cayzer S., Aickelin U, 2002. A Recommender System based on the Immune Network. In Proceedings of the Congress on Evolutionary Computation

Morrison T., Aickelin U, 2002. An Artifcial Immune System as a Recommender for Web Sites. In Proceedings of the 1st Conference on ARtifcial Immune Systems (ICARIS-2002). Canter-bury UK. 161-169

Maybury M., Greiff W., Boykin S., Ponte J, Mchenry C., Ferro L, 2004. Personalcasting: Tailored Broadcast News. Journal of User Modeling and User-Adapted Interaction vol. 14, Kluwer Academic Publishers, Netherlands 119-144.

Goren-Bar D, Glinansky O, 2004. FIT-recommending TV programs to family members. Journal of Computer and Graphics vol. 28, 149-156.

O' Sullivan D., Smyth B., Wilson D. C., Mcdonald K., Smeaton A, 2004. Improving the Quality of the Personalized Electronic Program Guide. Journal of User Modeling and User-Adapted Interaction vol. 14, Kluwer Academic Publishers, 5-36.

Virvou M, Savvopoulos A, Sotiropoulos D. N, Tsihrintzis G. A, 2007. Constructing Stereotypes for an Adaptive e-Shop using AIN-based Clustering, Lecture Notes In Computer Science, ICANNGA 2007, 11-14

Pessemier T.D., Deryckere T., Vanhecke K., Martens L., 2008. Proposed Architecture and Algorithm for Personalized Advertising on iDTV and Mobile Devices, In Journal of IEEE Transactions on Consumer Electronics.

Aragao V.R., Fernandes A.A., Carole A. 2001. Goble, Towards An Architecture For Personalization And Adaptivity In The Semanticweb. In Int. Conf. on Information Integration and Web-based Applications & Services, Linz, Austria, Sept.

Li Q., Kim B.M., 2004. Constructing User Profiles For Collaborative Recommender System, Lecture Notes in Computer Science, Springer-Verlag, pp 100-110.

Alspector J., Kołcz A., Karunanithi N., 1997. Feature-Based And Clique-Based User Models For Movie Selection: A Comparative Study. User Modeling and User-Adapted Interaction 7, 4 (Apr. 1997), 279-304.

Sotiropoulos D.N., Tsihrintzis G.A., Savvopoulos A., Virvou M., 2006. A Comparision of Customer Data Clustering Techniques in an e-Shopping Application, Adaptive Hypermedia and Adaptive Web-Based Systems, June 20 - 23, 2006

Rich E., 1979. User Modeling via Stereotypes, Journal of Cognitive Science, Vol. 3, 329-354

Brusilovsky, P., 2001. Adaptive Hypermedia, User Modeling and User-Adapted Interaction 11. Kluwer Academic Publishers 87-110

Chin, D. N. 2001. Empirical Evaluation of User Models and User-Adapted Systems. User Modeling and User-Adapted Interaction 11, 1-2 (Mar. 2001), 181-194.

Kobsa A: Generic User Modeling Systems. User Model. User-Adapt. Interact. 11(1-2): 49-63 (2001)

Kay J., Kummerfeld B., Lauder, P. 2002. Personis: A Server for User Models. In Proceedings of the Second international Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (May 29 - 31, 2002). Lecture Notes In Computer Science, vol. 2347. Springer-Verlag, London, 203-212.

Billsus, D., Brunk, C. A., Evans, C., Gladish, B., and Pazzani, M. 2002. Adaptive interfaces for ubiquitous web access. *Commun. ACM* 45, 5 (May. 2002), 34-38.