

# Hybrid Adaptation of Web-Based Systems User Interfaces

J. Sobecki

Department of Information Systems, Wrocław University of Technology  
Wyb. Wyspińskiego 27, 50-370 Wrocław, POLAND,  
sobecki@pwr.wroc.pl

**Abstract:** The interface adaptation is one of the methods for increasing the web-based system usability, especially when we consider differences among the population of users. We can distinguish demographic, content-based, and collaborative recommendations. The combination of these approaches that is called hybrid adaptation enables to overcome their disadvantages.

**Keywords:** Adaptive user interfaces, hybrid recommendation, web-based systems

## 1 Introduction

The success of the today's web-based information systems relies on the delivery of customized information for their users. The systems with this functionality are often called recommender systems [7]. We can distinguish three types of recommendations: demographic, content-based and collaborative. All of these three types use the user profiles to recommend particular items to the particular user. Recommender systems concerning user interfaces are usually called adaptive or personalized user interfaces [3] and partially belong to the more general class of intelligent user interfaces that contains also different interface agents [5].

One of the most important elements in the recommendation systems is the user model that is usually built of the two elements: the user data and the usage data [3]. The user data contains different information and it's the most common part is the demographic data containing: record data (name, address, e-mail, etc.), geographic data (zip-code, city, state, country), user's characteristics (sex, education, occupation), and some other customer qualifying data. The user data may contain also information about the users' knowledge, their skills and capabilities, their interests and preferences and also their plans and goals.

The second element of the user model, the usage data, is observed and recorded during the user's interactions with web-based systems. The usage data may concern selective operations that express users' interests, unfamiliarity or preferences, temporal viewing behavior, as well as ratings concerning the relevance of these elements.

The methods of building the user model are very differentiated. The user data is usually entered by the users themselves, but sometimes other methods from are used to associate these data with ones delivered for the other web-based system. The usage

data are usually collected automatically by the client and/or system sides by means of very different technologies ranging from CGI, PHP, ASP, Flash and Cookies to DoubleClick [14]. These entries can contain a lot of different data and settings, such as URL's of visited pages, links followed by the users, data entered by them into the forms on the pages and interface settings made by the users themselves as well as the goods purchased in the e-shops.

The user model is used by the recommendation process according to the implemented approach. Demographic filtering [7] takes descriptions of people from the user profile to learn the relationship between a particular item and the type of people who like it. Content-based filtering uses descriptions of the content of the items to learn the relationship between a single user and the description of the items according to the user usage description. Finally, the collaborative filtering uses the feedback from a set of some how similar people concerning a set of items in order to make recommendations.

In this paper, first the collaborative user interface recommendation with the demographic elements that is using consensus methods is presented. Then the hybrid architecture that is some combination of the former solution with some elements of content-based approach will be presented.

## 2 User Interface Recommendation Approaches

There are three basic recommendation approaches: demographic, content-based and collaborative. The demographic approach is using stereotype reasoning [3] in its recommendations and is based on the information stored in the user profile that contains different demographic features [7].

According to [3] stereotype reasoning is a classification problem that is aimed at generating initial predictions about the user. Usually the data used in the classification process is of the demographic type. This type of recommendation is also used in the initial steps of the collaborative user interface recommendations [10].

The demographic recommendations have however two basic disadvantages [7]:

- for many users recommendations may be too general;
- they do not provide any adaptation to user interests changing over time.

Content-based filtering takes descriptions of the content of the previously evaluated items to learn the relationship between a single user and the description of the new items [7]. The application of content-based approach enables personalized and effective recommendations for particular users, but has also some disadvantages:

- content-based approaches depends on so called objective description of the recommended items;
- it tends to overspecialize its recommendations;
- content-based approach is based only on the particular user relevance evaluations, but users usually are very reluctant to give them explicit, so usually other implicit, possibly less adequate, methods must be used.

The collaborative recommendations are able to deliver recommendations based on the relevance feedback from other similar users. Its main advantages over the content-based architecture are the following [7]:

- the community of users can deliver subjective data about items;
- collaborative filtering is able to offer novel items, even such that user have never seen before;
- collaborative recommendation utilizes item ratings of other users to find the best fitting one.

Collaborative recommended agents have also some disadvantages:

- when the number of other similar users is small then the prediction is rather poor;
- the quality of service for users of peculiar tests is also bad; this is rather difficult to get sufficient number of similar users to be able to make proper predictions;
- lack of transparency in the process of prediction and finally the user's personal dislike may be overcome by the number of other similar users opinions.

The disadvantages of content-based and collaborative recommendation could be overcome by applying the hybrid solution. In the works [10,13], where the concept and implementation of the collaborative user interface adaptation using consensus method is presented, the disadvantage of the insufficient number of the similar users at the early stages of the system operation was overcome by application of the demographic stereotype reasoning. However in this architecture the content-based recommendation is not implemented, but individual preferences of the interface settings that are selected manually by the user and stored in the interface profile and used in every system session.

### 3 User Profile

In works [10,12,13] we distinguish the user profile that contains user data (mainly demographic data) delivered by the user and the interface profile that is designated by the system and may be changed by the user during the personalization proces. In this paper, however, some changes to the previous model will be introduced. Like in the work [3] the user profile will contain the whole user model and the interface profile will be used only for the implementation of the current interface for the particular user.

The user data may contain different user characteristics: demographic data (name, sex, address, occupation, education, customer and psychographic data), user knowledge, user skills and user interests. The user data may also contain direct user preferences concerning the following elements of the user interface: the interface layout, the information content and its structure. For identification and authorization purposes data such as user login and password are also included in the user profile.

The user profile could also contain the usage data that may be observed directly from the user's interaction with a web-based system. This part should contain all versions of interface settings, visited pages and if appropriate also other information on other actions concerning the system purposes, such as purchase of goods, order of catalogs, request for some attendance, etc. In order to know whether the interface settings are appropriate for the particular user we should also store the usability value associated with these settings [8].

In order to find similar users by means of clustering methods and find consensus among interface settings [10] it is necessary to define the distance function in the space of the interface profile.

### 3.1 Formal Description of the User Profile

In the recommender systems the user profile could be represented by many different forms: binary vectors, feature vectors, trees, decision trees, semantic networks. In our paper we propose to describe the user profile as a tuple defined in the following way: the finite set  $A^u$  of attributes of the user profile and the set  $V^u$  contains attribute values, where:  $V^u = \bigcup_{a \in A^u} V_a^u$  ( $V_a^u$  is the domain of attribute  $a$ ). The tuple  $p$  is a

function  $p : A^u \rightarrow V^u$  where  $(\forall a \in A^u)(p(a) \in V_a^u)$ . This type of representation is also called as a single valued information system. It is also possible to consider so-called multivalued information that enables to be the attribute values not only atomic but also sets of values. system. In this case we introduce  $\Pi(V_a^u)$  that denote the set of subsets of set  $V_a^u$  and  $\Pi(V^u) = \bigcup_{a \in A^u} \Pi(V_a^u)$ . The interface profile is represented by a

tuple that is a function  $r : A^i \rightarrow \Pi(V^i)$  where  $(\forall a \in A^i)(p(a) \in \Pi(V_a^i))$ . We can also consider more general representation that means the ordered set or even ordered set with repetitions.

The user profile may be represented by other means. An example of application of the tree structure for the user interface description can be found in work [12].

In the user profile we can distinguish several parts represented by the attribute sets. We can distinguish the demographic attributes set  $D$ , the set of the interface setting attributes  $I$ , the set  $C$  of attributes associated with the content (for example visited pages, purchased or ordered items, retrieved elements) and the set  $U$  of the usability or relevance of the interface or contents. Finally, we can distinguish some attributes used for identification and authorization purposes  $T$ .

### 3.2 Distance Function Definition

The distance function between values of each attribute of the user profiles is defined as a function  $\delta^{at} : V_a^u \times V_a^u \rightarrow [0,1]$  for all  $a \in B \subseteq A^u$ . This function should be given by the system designer and fulfill all the distance function conditions but not especially all the metrics conditions. The distance function values could be enumerated or given in any procedural form.

The distance between elements of the user profiles could be defined in many different ways. First, the distance between tuples  $i$  and  $j$  could be defined as a simple sum of distances between values of each attribute:

$$\delta(p_i, p_j) = \sum_{a \in B} \delta^{at}(p_i(a), p_j(a))$$

or we can consider the cosine distance or define the distance as a square root of sum of squares of distances (Euclidean distance). We can also indicate the importance of

each attribute  $a$  by multiplying the distance by appropriate factor defined as a function  $c : A \rightarrow [0,1]$ :

$$\delta(p_i, p_j) = \sum_{a \in B} [c(a) * \delta^{at}(p_i(a), p_j(a))]$$

The functions shown above are devoted for the user profile representation in form of a tuple, but in the case of the tree structures we must consider other distance functions [12].

### 3.3 User Clustering Based on the User Profile

Clustering problem could be defined as a partition of the given set of user profiles into subsets such that a specific criterion is optimized. The criterion is often defined as the average squared Euclidean distance between a profile and the corresponding cluster center. To minimize this criterion we can use k-means clustering that partitions the set of the profiles into  $k$  non-overlapping clusters that are identified by their centers. This problem is known to be NP-hard, but it is still attractive because of its simplicity and flexibility [2]. It has however some disadvantages that reveals especially in case of a large datasets, these are: its low speed and lack of scalability; it is possible to obtain local minima instead of global ones.

In the interface recommendation implementation described in [13] Dattola clustering algorithm [1] that is known from the field of Information Retrieval was used. This algorithm is not NP-hard but produces the sub-optimal solution to the clustering problem.

In the Datolla algorithm first the initial centroids must be selected, in our case they are selected by experts. Then for each user profile the distance function (see previous section) between the profile and each centroid is determined. The profile is joined to the group with the closest the centroid and also lower that assumed threshold, those above are assigned to the class of so-called isolated elements. Then for each group the centroides are recalculated and the process is repeated until no one profile changes its class assignment. Finally all profiles from the class of isolated elements are assigned to the group with the lowest distance function values or left as a separate group.

In the field of data mining however other algorithm is used to solve k-means problem. Its name is Lloyd's algorithm [2] and its steps are following. First, select randomly  $k$  elements as the starting centers of the clusters (centroides). Second, assign each element of the set to a cluster according to the smallest distance to its centroid. Third, recompute the centroid of each cluster, for example the average of the cluster's elements. Fourth, repeat steps 2 and 3 until some convergence conditions have not been met (for example centroides do not change).

The attractiveness of this algorithm lies in its simplicity and its ability to terminate when using the above mentioned convergence condition and for configurations without equidistant elements to more than one centroid. There is, however, one important problem with k-means algorithm, namely the algorithm takes a long time to run. First, the step 2 that has to be performed in each iteration costs  $O(kdN)$ , where  $d$  is the dimension of each element and  $N$  is the number of elements. Second, algorithm usually needs many iterations to terminate. There are however quite many modification of this algorithm that run faster, for example bisecting k-means [2].

### 3.4 Interface Profile Determination by Means of Consensus Methods

The consensus methods are used to find collaborative interface recommendation for particular user. In order to do this we must have the group of similar users  $G$ . Their interface settings with assigned usability values are used for the consensus determination. Let  $j$  be the index of the a group  $G$  member,  $r_j$  be the tuple that is sub-tuple of the  $p_j$  such that  $r_j(a)=p_j(a)$  for all  $a \in C$  where  $C \subset A$  is a set of the interface setting attribute and  $u(j)$  usability measure assigned to the each of the interface users. Let assume that the usability measure falls in the  $[0,1]$  interval and the value 0 denotes completely useless interfaces and 1 denotes ideally useful interface. Then to find the consensus (recommended interface for particular group of users  $G$ ) we must find the interface settings  $r$  that conform the following formula [12]:

$$\min(\sum_{j \in G} u(j) * \delta(r, r_j))$$

This problem could be computationally difficult, but according to [9] we can reduce the computation by finding the minimal value for all attributes  $a$  of a tuple separately:

$$\min(\sum_{j \in G} u(j) * \delta(r(a), r_j(a)))$$

## 4 Hybrid Recommendation System Architecture

In works [10,13] the user interface recommendation for the web-based system is based on demographic recommendation and collaborative recommendation is presented. The system designers prepare centroides that represent stereotype classes of users. Each centroid has the associated interface profile that for the designers represents the most efficient interface settings for these classes. At start each new user registers to the system. The user delivers mainly demographic data (value of the attributes from the set  $D$ ) and identification data (set  $T$ ) that is stored in user data part of the user profile. Then the class centroid with the closest distance to the user profile is found and associated interface profile offered to the user.

According to the interface profile the actual user interface content, layout and structure is generated. The user may start to work with the system and if he or she wishes also modify the interface settings (values of the attributes from the set  $I$ ). Finally, these settings together with usability evaluation (set  $U$ ) given by user are also stored in the user profile.

When the system registers required number of users, first users are clustered (using attribute values from the set  $D$ ) and then according to these clusters using consensus methods on values stored in the set  $I$ , new centroides and associated interface profiles for recommendation are distinguished. These procedures may be repeated from time to time in the following occasions: many new users registering to the system or interface recommendations become poor usability ratings.

In the hybrid recommendation architecture we can introduce some modifications concerning individual relevance rating of the interface elements content recommendation. First, using the stereotype reasoning we can ascribe for specified values of the attributes  $D$  appropriate values of the attributes from the set  $I$  (i.e. when the user specifies the country in the demographic data we can recommend the

language of the system or we can recommend the weather forecast concerning the user's place of living). Second, the personalization settings from the set  $I$  made by the user and those recommended by the system and presented to the user are not the subject of any further automatic adaptation without the direct user permission.

Finally, we can implement the content-based recommendation. We must remember however that the rules for efficient content-based recommendations strongly depend on the goals of the web based system. For example for web-based information retrieval systems we can consider the previous relevant items as a basis for recommendation of further retrievals. In this case many different methods can be used: fuzzy retrieval, Bayesian networks or other intelligent information retrieval method. The same methods we can use for other similar tasks such as news and e-mails filtering or spam detection. Those methods were especially applied in the field of the interface agents, for example in Letizia [5] or Apt Decisions [11].

For quite many systems however, the logic used for the retrieval systems does not hold, for example when a user buys a thing that is needed in only one item (e.g. car, TV set, DVD player, etc.). In that cases we should rather recommend items that other users who bought the same item were also interested in, for example insurance when a user buys a car. In many cases the logic of the system may be even more complicated, so for each recommended item, we shall define precise relationship between the user profile (or also other users profiles) and this item.

In the content item recommendation we can also consider application of the design heuristics concerning the content and presentation of the information:

- meaning: the information has meaning when it concerns many people;
- time: the information is delivered on time;
- publicity: the information concerns a very well known person or organization;
- adjacency: the information is adjacent when it concerns the user residence place;
- conflict: the information concerns the conflict among people;
- peculiarity: the information is interesting when it is quite unusual;
- timeliness: the information is interesting when it concerns current event.

## 5 Summary

Recent works on several implementations of consensus based collaborative user interface recommendations [10] have proven that it is possible to implement this method for a range of different web-based information systems. All of the implemented systems were used in the field of the product or services promotion. The system concerning a particular car model were analyzed in more details [13], the results prove that collaborative user interface adaptation leads to usability improvements. The other systems, however, are still investigated and we have no the final result yet.

The hybrid approach was up till now only partially implemented but comparing to the pure collaborative approach, the experiments requires longer times of users work with the system (also in several separate sessions) to gather sufficient data for content-based recommendation.

## References

1. Dattola RT (1968) A fast algorithm for automatic classification. In: Report ISR-14 to the National Science Foundation, Section V, Cornell University, Dep. of Computer Science.
2. Kanungo T, Mount DM, Netanyahu, NS, Piatko C, Silverman R, Wu AY (2002) An Efficient k-means clustering algorithm: analysis and implementation. *IEEE Tran. On Pattern Analysis And Machine Intelligence* 24(7): 881-892
3. Kobsa A, Koenemann J, Pohl W (2001) Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships. *Knowledge Eng. Rev.* 16(2): 111-155.
4. Langley P (1999) User Modelling in Adaptive Interfaces. *Proceedings of the Seventh International Conference on User Modeling*: 357-371
5. Lieberman H (1997) Autonomous Interface Agents. *Proc. CHI 97, ACM*: 67-74.
6. Mobasher B, Cooley R, Srivastava J (1999) Automatic personalization based on Web usage mining. Technical Report TR99010, Dep. of Computer Science, DePaul University
7. Montaner M, Lopez B, de la Rosa JP (2003) A Taxonomy of Recommender Agents on the Internet. *Artificial Intelligence Review* 19: 285-330.
8. Newman WM, Lamming MG (1996) *Interactive System Design*. Addison-Wesley, Harlow.
9. Nguyen NT (2001) Conflict Profiles' Susceptibility to Consensus in Consensus Systems. *Bulletin of International Rough Sets Society* 5(1/2): 217-224
10. Nguyen NT, Sobecki J (2003) Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems. *Universal Access in Inf. Society* 2(4): 342-358
11. Shearin S, Lieberman H (2001) Intelligent Profiling by Example. In: *Proc of the Conf on Intelligent User Interfaces*, ACM Press, 2001.
12. Sobecki J (2003) XML-based Interface Model for Socially Adaptive Web-Based Systems User Interfaces. *Lecture Notes in Computer Science* 2660: 592-598
13. Sobecki J, Weihberg M (2004) Consensus-based Adaptive User Interface Implementation in the Product Promotion. To be published as a chapter in book "Design for a more inclusive world", by Springer-Verlag (London).
14. Whalen D (2002) The Unofficial Cookie FAQ, Version 2.54 Contributed to Cookie Central by David Whalen. <http://www.cookiecentral.com/faq/#2.7>.