Cross-Technique Mediation of User Models

Shlomo Berkovsky[1], Tsvi Kuflik[1], and Francesco Ricci[2]

[1] University of Haifa, Haifa
slavax@cs.haifa.ac.il, tsvikak@is.haifa.ac.il
[2] ITC-irst, Trento
ricci@itc.it

Abstract. Nowadays, personalization is considered a powerful approach for designing more precise and easy to use information search and recommendation tools. Since the quality of the personalization provided depends on the accuracy of the user models (UMs) managed by the system, it would be beneficial enriching these models through mediating partial UMs, built by other services. This paper proposes a cross-technique mediation of the UMs from collaborative to content-based services. According to this approach, content-based recommendations are built for the target users having no content-based user model, knowing his collaborative-based user model only. Experimental evaluation conducted in the domain of movies, shows that for small UMs, the personalization provided using the mediated content-based UMs outperforms the personalization provided using the original collaborative UMs.

1 Introduction

The quantity of information available on the Web grows rapidly and exceeds our limited processing capabilities. As a result, there is a pressing need for intelligent systems providing personalized services according to user's needs and interests, and delivering tailored information in a way most appropriate to the user [10]. Providing personalized services to the users requires modeling their preferences, interests and needs. This data is referred in the literature as a User Model (*UM*) [8].

Typically, service providers build and maintain proprietary UMs, tailored to the application domain of the service and to the specific personalization technique being exploited. Since the accuracy of the provided personalized service heavily depends on the characteristics and quality of the UMs, different services would benefit from enriching their UMs through importing, translating and aggregating partial UMs, i.e., UMs built by other, possibly related, services. This can be achieved through *mediation* of partial UMs [2].

The main functionality of UM *mediator* [2] is to acquire partial UMs built by other service providers, and to aggregate the acquired UMs into a UM for the target service. Analysis of the state-of-the-art personalization techniques and application domains yields four groups of services that can potentially provide valuable partial UMs for building a UM for a service from domain *d* exploiting technique *t*: (1) services from *d* that also exploit *t*, (2) services from *d* that exploit another technique *t'*, (3) services

V. Wade, H. Ashman, and B. Smyth (Eds.): AH 2006, LNCS 4018, pp. 21 – 30, 2006.

from another, relatively similar, domain d' that also exploit t, and (4) services from another, relatively similar, domain d' that exploit another technique t'.

Clearly, for the first group of services, the mediation of partial UMs is quite simple, as both the content and the representation of the UMs are similar. Although the mediation should still cope with semantic heterogeneity of the UMs, e.g., synonyms or multilinguality, this can be resolved through adapting the solutions proposed by the Data Integration community [3]. This is not the case for the second and third group of services. Mediation of the UMs, whether represented according to a different personalization technique, or representing a different application domain, requires identifying the relationships between the knowledge modeled by the source UMs and the knowledge required by the target UM. This can be achieved through exploiting a rich semantic *Knowledge Base* (KB), covering both the target and the source UMs, which will actually facilitate the translation of the 'overlapping' (i.e., stored by the source and needed by the target) parts of the UMs. The above two types of mediation will be referred to as *cross-technique* and *cross-domain* mediations[1], respectively.

This paper focuses on cross-technique mediation of partial UMs from collaborative filtering recommender systems, where a vector of explicit ratings on a set of objects is provided by a user [5], to a content-based UM, represented as a list of the user's preferences [9]. The mediation exploits a KB facilitating the identification of commonalities in positively or negatively rated objects, as derived by the collaborative filtering UM, and generalizing them into a weighted list of topics liked/disliked by the user.

The proposed mediation mechanism was implemented, and its accuracy was evaluated using EachMovie, a publicly available movie ratings dataset. IMDb database (The Internet Movie Database, *http://www.imdb.com*) was exploited for extracting the features of the rated movies, such as genre, actors, directors etc. Then, the UMs mediation was accomplished through translating the collaborative ratings, of the user to whom a recommendation is to be provided, into a weighted list of liked/disliked features. The translation was based on the assumption that user's rating on a movie steadily reflect her/his preferences of the features of the movies, such as preferred genre, or director. The generated UMs served as a basis for generating content-based predictions. Two experiments were performed. The first was designed to fine-tune and optimize the predictions generation mechanism, while the second actually evaluated the accuracy of the predictions using the well-known Mean Average Error (MAE) metric [6]. Experimental results demonstrate high accuracy of the generated predictions, validating usefulness of the collaborative to content-based translation, and demonstrating the applicability of cross-technique mediation of UMs.

The rest of the paper is organized as follows. Section 2 briefly presents prior research efforts on mediation and aggregation of UMs. Section 3 describes the proposed approach for cross-technique UM mediation and elaborates on translation of collaborative filtering UMs to content-based UMs. Section 4 presents and discusses the experimental results, and section 5 concludes and presents future research topics.

---

[1] Note that currently our research does not deal yet with a combination of cross-technique and cross-domain mediations, since this would require multiple translations of partial UMs, which may 'contaminate' the original data.

2 Mediation and Aggregation of User Models

Centralized generation of the UMs, as a composition of partial UMs stored by different personalization services, is proposed in [7]. To accomplish this, each service maintains a mechanism for extracting the relevant parts of the central UM, and updating the central UM after the service is provided. A similar approach is discussed in [11], proposing to use Unified User Context Model (UUCM) for improving the partial UMs built by individual services. To provide personalization, services extract the required data from the UUCM, deliver the service, and update the UUCM. However, the centrality of the UM poses a severe privacy problem that should be resolved.

In recommender systems, many prior works on *hybrid* recommender systems tried to integrate multiple techniques in the prediction generation process [4]. Hybrid recommenders usually combine two or more techniques to improve predictions accuracy, but they are not concerned with the conversion of UMs between the techniques. Other related approach is presented in [12], that integrates collaborative and content-based techniques by basing collaborative-based similarity assessments on the content-based UMs. In [1], the authors extract content-based UMs from collaborative UMs and use both of them for the purposes of the predictions generation. Conversely, the current work focuses on generation of pure content-based predictions, based solely on the UM provided by the mediator. As such, it can not be classified as a hybrid one.

3 Collaborative Filtering to Content-Based Translation of UMs

Collaborative filtering is probably one of the most popular recommendation techniques. It recognizes cross-user correlations and generates predictions by weighing the opinions of similar users [5]. The input for the collaborative filtering is a matrix of users' ratings on a set of items, where each row represents ratings of a single user and each column represents ratings on a single item. Thus, collaborative filtering UMs are represented as ratings vectors $UM_{CF}=\{i_1:r_1,\ i_2:r_2,\ \ldots,\ i_n:r_n\}$, where every pair $i_k:r_k$, corresponds to a real rating r_k provided by the user on an item i_k.

Content-based filtering [9] builds personalized recommendations by taking as input: (1) the features of items that have been rated by the user, and (2) the set C of available items, not yet rated by the user, i.e., the candidate recommendations. The output recommendation is a subset of C, containing items whose features match the features of items which were preferred by the user. Content-based recommenders generate recommendations based on the set of features weighed according to a predefined scale, such as like/dislike or a number between 0 and 1. Thus, content-based UMs are represented as a list $UM_{CB}=\{f_1:w_1,\ f_2:w_2,\ \ldots,\ f_n:w_n\}$, where f_k denotes one of the domain features and w_k the level of the user's preference regarding this feature.

For instance, a collaborative UM for movie recommendations is a collection of movies and their respective ratings, explicitly provided by the user. Consider the following sample UM_{CF}=*{"The Lord of The Rings":1, "The Matrix":0.8, "Psycho":0.2, "Friday the 13th":0, "Star Wars":0.9, "The Nightmare":0.1", "Alien":0.9}*, built on a continuous scale of ratings between 0 to 1. Although a collaborative UM represents the user as a set of ratings, it can be recognized that the user likes science-fiction movies, and dislikes horror movies. Thus, content-based UM of the user may be similar to UM_{CB}=*{science-fiction:0.9, horror:0.1}*, where the genre weights are computed as an average of the ratings given to the movies in this genre. Similarly to the genre weights, also the weights of other features, such as, directors and actors can be computed.

To handle the translation of collaborative UMs into content-based UMs, a rich movies' KB is needed for identifying the content of the movies, and providing the required lists of genres, actors, directors, and so forth. In this work, an offline version of the IMDb movie database (*http://www.imdb.com*) served as the translation KB. IMDb provides information in *49* feature categories, such as genre, actors, directors, writers, cinematographers, composers, keywords, languages, etc. For the sake of simplicity, only 7 feature categories were used in this work: *genres*, *keywords*, *actors*, *actresses*, *directors*, *production countries* and *languages*, as these categories seem to most affect the user's decision in selecting a movie.

Translating collaborative UMs to content based UMs takes the user's ratings vector as an input. Since different users may express their ratings in different ways (e.g., rating *4*, provided by a user whose average rating is *2* should be treated differently than rating *4* provided by a user whose average is *3.5*), users' ratings were normalized in order to eliminate individual differences between users. This was done by subtracting the average rating of the user from the provided ratings.

For each movie rating in a collaborative UM, a list of a movie's features (in the above categories) was extracted from IMDb. The weights of the features were updated according to the normalized rating of the movie, provided by the collaborative vector. In other words, the normalized rating of the movie was added to the weights of all the movie genres, actors and directors involved in the movie (and similarly for all the remaining categories). In addition, the number of occurrences for each feature, i.e., the number of movies rated by the user and having that feature was recorded.

For example, consider the rating *"Star Wars":0.9*, given by a user whose average rating is *0.6*. According to the IMDb, the genres of *"Star Wars"* are *action*, *adventure*, *fantasy* and *science-fiction*. Thus, the existing weights of these four features are increased by *0.3*. Similarly, the weights of the movie director *George Lucas*, and all the actors involved in the movie are increased by *0.3*. The number of occurrences for the above genres, *George Lucas*, all the actors and other features is increased by one.

After the content-based UM is generated, the user is modeled as a set of weights $\{w_{i(1)}, \ldots, w_{i(k)}\}$ for a subset of size k (depending on the user) features available in the *7* categories, and corresponding feature frequencies $\{c_{i(1)}, \ldots, c_{i(k)}\}$. Hence, a predicted rating for a movie m can be generated by extracting from IMDb all the relevant features of m and computing the prediction as a weighted average of the weights of the features that are both in the UM and in the movie description:

$$rating(m) = \frac{\sum_{j \in F(u) \cap F(m)} w_j c_j}{\sum_{j \in F(u) \cap F(m)} c_j}$$

where $F(u)$ are the features in the user model and $F(m)$ are the features in the movie model. Finally, a movie with the highest prediction is recommended to the user.

Note that the predictions are generated solely based on content-based UM, which is derived from the collaborative UM. As such, the predictions mechanism is capable of building content-based predictions regardless of the number of ratings available for the given movie. Therefore, this approach resolves the well-known *first-rater problem* in collaborative filtering [5], where an item cannot be recommended unless it was already rated by a sufficient number of users. Nevertheless, as a pure content-based recommender, it may suffer from an inherent *serendipity problem*, i.e., it can recommend only movies that are similar to the movies already rated by the user.

3.1 Fine-Tuning of the Prediction Mechanism

Although the proposed mechanism is capable of generating predictions regardless of the number of available ratings on a movie, it may suffer from instability (i.e., undesired fluctuations affected by minor factors). Since IMDb contains a lot of information for each movie, content-based UMs built from collaborative UMs containing a dozen of ratings only include thousands of features of actors, actresses and keywords occurring only once. This is explained by the fact that hundreds of actors and actresses are involved in every movie, while the number of genres or directors is at most *3-4*. As the UM accumulates movie data, the number of such *once-occurring* features increases, and they add noise to the prediction mechanism by becoming a dominant factor and 'blurring' the important features. In addition to once-occurring features, content-based UMs typically store a large number of *neutral* features, i.e., features to which the user is indifferent, which are sometimes rated positively and sometimes negatively. As a result, their weight is close to *0*, regardless of the number of occurrences in the UM. Similarly to once-occurring features, a large number of neutral features also adds noise to the prediction mechanism by 'blurring' the *differentiating* features.

To filter the influence of once-occurring and neutral features, two thresholds were defined: (1) *min-occurs* – minimal number of occurrences for a feature, and (2) *confidence* – minimal weight of a feature. The prediction mechanism was modified to take into account only those features, that occur at least *min-occurs* times, and whose weight is above *confidence* or below *-confidence* threshold. However, the weight of a feature depends on the number of occurrences of the feature. Thus, a *normalized weight* of the features was computed by dividing the weight of a feature in the content-based UM by the number of occurrences of that feature. The following pseudo-code describes the fine-tuned recommendation generation process:

```
Recommend (Content-Based-UM u, set-of-movies M)
  foreach m∈M
     retrieve F(m) = set-of-feature of m
     for each j∈F(m)
        if j∈F(u) AND |norm-w_j|>confidence AND c_j>min-occurs
           take j into account for prediction of rating(m)
     compute rating(m)
  return m with maximal predicted rating(m)
```

We note that the proposed prediction mechanism assigns equal weights for features across different categories, i.e., there is no additional weighing factor that reflects the importance of a category. Although the weighing issues are important, they fall beyond the scope of the current work.

4 Experimental Results

The above collaborative to content-based translation was tested over publicly available EachMovie dataset (*http://research.compaq.com/SRC/eachmovie/*). EachMovie is a collaborative filtering dataset, storing *2,811,983* ratings between *0* and *1* of *72,916* users on *1,628* movies. In our experiments, we selected a subset of *1,529* movies, which were identified in IMDb, and *47,988* users whose variance of ratings is not *0* (i.e., the ratings are not identical) that rated more than *10* movies. Thus, a total number of *2,667,605* ratings were obtained, producing a sparse dataset with a sparsity of *3.64%*. Most of the users in the dataset rated relatively few movies. Table 1 shows the distribution of the number of rated movies among the users:

Table 1. Distribution of ratings among the users in the dataset

rated movies	0 to 25	26 to 50	51 to 75	76 to 100	101 to 125	126 to 150	151 to 175	176 to 200	201 to 225	226 to 250	251 to 300	301 to 500	over 500
number of users	17,321	13,788	6,514	3,609	2,302	1,349	887	609	441	327	358	436	47

The first set of experiments was designed to fine-tune the prediction mechanism by selecting the most appropriate values for the *confidence* and *min-occurs* thresholds. To accomplish this, one of the thresholds was set to a constant, while the values of the second were gradually modified. For each value of the modified threshold, a subset of *1,000* users that rated at least *100* movies was selected, and for each one of them, *90%* of the ratings were defined as the training set and the remaining *10%* as the test set. Then, the collaborative UM was translated to the content-based UM, using only the ratings contained in the training part, and predictions for the movies in the test set were built according to the above prediction mechanism. Accuracy of the predictions using the given threshold values was evaluated by the well-known MAE metric [6].

To find the most appropriate value of *confidence*, *min-occurs* threshold was set to *min-occurs=2* for all the categories, and the values of *confidence* threshold were gradually increased from *0* to *0.5*. To provide an initial indication for different relative importance of different categories, the predictions were generated in two ways: (1) using features from all *7* categories, and (2) using features from all the categories, except *keywords*. We note that a high *confidence* threshold reduces the number of features in a UM and therefore ratings of some movies cannot be predicted. Hence, for each value of *confidence*, the prediction rate (i.e., the percentage of movies whose ratings were predicted) was computed. Figure 1 illustrates the results of the experiments. The horizontal axis shows the values of *confidence* threshold, and the vertical – the MAE and prediction rate values. The dotted curves show the prediction rate values, while the continuous ones the MAE. The dark curves show the results based on 7 categories, while the light curves are based on 6 categories, excluding *keywords*.

As can be seen, MAE values initially slightly decrease with the *confidence*, and then monotonically increase. This is explained by the influence of neutral features. If the *confidence* threshold is low, and neutral features are not filtered, they add noise to

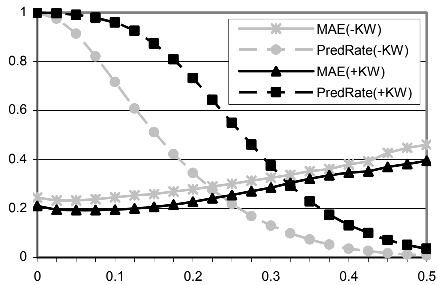

Fig. 1. MAE and prediction rate vs. *confidence* threshold

the prediction mechanism and the MAE is higher. When the *confidence* increases, neutral features are filtered and MAE decreases. However, high values of the *confidence* filter also differentiating features, and MAE increases again. Thus, *confidence=0.025* was chosen as an optimal value, where the MAE is minimal and prediction rate is high (over *0.99*). Prediction rate monotonically decreases with *confidence*, since when more features are filtered, the task of generating a prediction is harder to accomplish. Note the difference between the experiments including and excluding the *keywords* features in prediction generation. Both metrics of MAE and prediction rate show that it is beneficial to take the keywords into account.

After determining the value of *confidence* threshold it was used for choosing the optimal value of the *min-occurs* threshold. Considering *min-occurs*, we observed two different situations corresponding to two types of categories. For the first one, such as *genres* or *languages*, the number of possible features is low. As a result, the *min-occurs* threshold is relatively high. For the second, such as *actors* or *keywords*, the number of possible features is very high, and the *min-occurs* threshold is low. The categories were separated, and the same methodology was used to determine the optimal *min-occurs* value for each category. The value of the *confidence* threshold was set to *0.025*, and the values of the *min-occurs* thresholds were gradually modified to determine the optimal threshold. Note that for each category, a separate experiment was conducted where the predictions were generated based only on the features of this category, and MAE and prediction rate values were computed as a function of the *min-occurs* threshold. The experiment was conducted for the same *1,000* users that rated at least *100* movies. Due to a lack of space, figure 2 illustrates the results of the experiments for two representative categories: *genres* (left) and *keywords* (right). In both experiments, the horizontal axis shows the percentage of the rated movies containing the given feature and the vertical – the MAE and prediction rate values.

The results show that for the *genres* category, MAE monotonically increases with *min-occurs*. Thus, filtering of *genres* features hampers the accuracy of the generated predictions, and practically, any feature from this category is valuable. This means that the optimal *min-occurs* threshold for the *genres* category is 0. Conversely, the *keywords* MAE curve behaves similarly to the *confidence* curve. It initially decreases with *min-occurs*, filtering the noisy features, and then monotonically increases, as for a higher *min-occurs* threshold, also important features with a high number of occurrences are being filtered. As for the prediction rate, it monotonically decreases with *min-occurs*. Similarly to the *confidence* threshold, this is explained by the fact that the high threshold filters important features, and the prediction generations are harder.

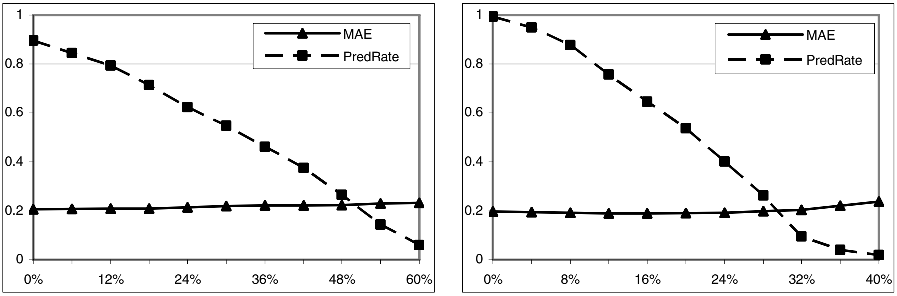

Fig. 2. MAE and prediction rate vs. *min-occurs* threshold for *genres* (left) and *keywords* (right) categories

Similar behavior was also observed for other categories. For categories with a small number of possible features, such as *production countries* and *languages*, any filtering hampers the MAE, and therefore, the optimal *min-occurs* threshold is *min-occurs=0*. For categories with a large number of features, such as *actors*, *actresses* and *directors*, initial filtering improves the MAE, whereas additional increase of *min-occurs* threshold causes the MAE to monotonically increase. The following table summarizes the optimal values of the *min-occurs* threshold for different categories:

Table 2. Values of min-occurs threshold for features from different categories

category	*genres*	*keywords*	*actors*	*actresses*	*directors*	*countries*	*languages*
min-occurs (%)	0	12	2	1.2	0.45	0	0

The determined *min-occurs* and *confidence* thresholds were applied in the second set of experiments, designed to compare the original collaborative and content-based recommendations. In principle, the collaborative and content-based recommenders are designed to recommend different types of movies. A collaborative recommender will recommend movies rated positively by similar users, while a content-based – movies similar to the movies that were rated highly by the user. Thus, the best experiment would be generating *sets* of recommended movies and conducting user studies evaluating these sets. Since we were unable to conduct such experiments, the accuracy of the generated predictions was compared using the MAE metric [6].

For this experiment, the users in the dataset were again partitioned into *12* groups of users, according to the number of rated movies[2]. *325* users were selected from each group, and the collaborative UM of each selected user was partitioned to *90%* training set and *10%* test set. Then, two types of predictions were generated: collaborative predictions based on the collaborative training set UM, and content-based predictions based on the translated content-based UM. For each group, collaborative and content-based

[2] In the first experiment, we selected *1,000* users that rated at least *100* movies. For the second experiment, we defined *12* groups of *325* users each, a total of *3,900* users. Although there is overlapping, it is partial, and it is only for groups of users with over *100* rated movies.

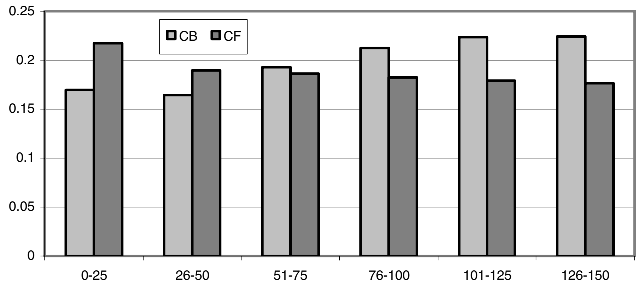

Fig. 3. MAE of content-based (left, light columns) and collaborative (right, dark columns) prediction vs. the number of rated movies in the UM

MAE values were computed. Figure 3 shows the MAE values. The horizontal axis reflects the number of users in a group, while the vertical axis stands for the MAE. Due to the lack of space, MAE values of the first *6* groups only are shown.

The chart shows that the MAE of content-based predictions for the UMs containing below *50* movies is relatively low, approximately *0.17*. This is explained by the observation that for a low number of rated movies in the UM, it is easy to find the weights of differentiating content-based features, while the number of neutral features is still low, and they do not dominate in the predictions generation. For larger UMs, between *50* and *100* movies, the MAE increases with the number of rated movies. We conjecture that this happens due to a larger number of neutral features, which hamper the accuracy of the generated prediction. Finally, for UMs with over *100* rated movies, the MAE stabilizes at approximately *0.22*. For most of the groups, the prediction rate is over *0.99* (except the group of less than *25* movies, where it is *0.974*). This means that predictions can be computed for almost every movie.

Comparison of the content-based and collaborative MAE values shows that for below *50* rated movies in the UM, pure content-based prediction based on the translated artificial UMs outperforms collaborative predictions, based on the original UMs. According to table 1, *64.8%* of the users in the dataset rated up to *50* movies. Thus, improving the predictions accuracy in this range is extremely important. Since the accuracy of the collaborative predictions for this size of the UMs is quite low, translation of the UMs and further content-based predictions provide a solid alternative technique. For a larger number of rated movies in the UMs, collaborative predictions outperform the content-based ones. However, the difference in the MAE is smaller than *0.05*, which indicates a reasonable performance of content-based predictions. We conjecture that weighing categories and specific features may significantly improve the accuracy of content-based predictions also for larger UMs.

5 Conclusions and Future Research

This work presents cross-technique mediation of UMs and demonstrates the feasibility of translating from collaborative to content-based UMs, allowing a content-based recommender to generate recommendations for a new user, whose UM was imported from a collaborative recommender. The experimental study first focused on determining the thresholds which filter out irrelevant and neutral features. Then, the thresholds were applied and the accuracy of the generated content-based predictions was evaluated and compared to the accuracy of the original collaborative predictions. The experiments showed that for a small number of rated movies in the UMs (typical for most users), the accuracy of content-based predictions is higher than that of collaborative-based prediction. This leads to the conclusion that cross-technique mediation of the UMs is feasible, and can also improve the quality of the personalization provided.

The discussed prediction mechanism is quite simple, as it assigns equal weights to different categories of the UM data. In the future, we plan to exploit various learning techniques to infer the weights of the categories and specific features within the categories. We believe this will significantly improve the accuracy of the personalization provided and strengthen the proposed cross-technique mediation. We also plan to extensively evaluate the proposed approach for other cross-technique mediations (e.g., the reverse translation, from content-based to collaborative UMs) and in different application domains.

References[3]

[1] C.Basu, H.Hirsh, W.Cohen, "*Recommendation as Classification: Using Social and Content-Based Information in Recommendation*", in proc. of the AAAI Conference, 1998.

[2] S.Berkovsky, "*Decentralized Mediation of User Models for a Better Personalization*", in proc. of the AH Conference, 2006.

[3] P.A.Bernstein, S.Melnik, "*Meta Data Management*", in proc. of the ICDE Conference, 2004.

[4] R.Burke, "*Hybrid Recommender Systems: Survey and Experiments*", in User Modeling and User-Adapted Interaction, vol.12(4), p.331-370, 2002.

[5] J.L.Herlocker, J.A.Konstan, A.Borchers, J.Riedl, "*An Algorithmic Framework for Performing Collaborative Filtering*", in proc. of the SIGIR Conference, 1999.

[6] J.L.Herlocker, J.A.Konstan, L.G.Terveen, J.T.Riedl, "*Evaluating Collaborative Filtering Recommender Systems*", in ACM Trans. on Information Systems, vol.22(1), p.5-53, 2004.

[7] J.Kay, B.Kummerfeld, P.Lauder, "*Managing Private User Models and Shared Personas*", in proc. of the UbiUM Workshop, 2003.

[8] A.Kobsa, "*Generic User Modeling Systems*", in User Modeling and User-Adapted Interaction, vol.11(1-2), p.49-63, 2001.

[9] M.Morita, Y.Shinoda, "*Information Filtering Based on User Behavior Analysis and Best Match Retrieval*", in proc. of the SIGIR Conference, 1994.

[10] M.D.Mulvenna, S.S.Anand, A.G.Buchner, "*Personalization on the Net Using Web Mining*", in Communications of the ACM, vol.43(8), p. 123-125, 2000.

[11] C.Niederee, A.Stewart, B.Mehta, M.Hemmje, "*A Multi-Dimensional, Unified User Model for Cross-System Personalization*", in proc. of the E4PIA Workshop, 2004.

[12] M.J.Pazzani, "*A Framework for Collaborative, Content-Based and Demographic Filtering*", in Artificial Intelligence Review, vol.13(5-6), p.393-408, 1999.

[3] The authors would like to thank Boris Bolshem and Sveta Ogiyenko for their assistance with the implementation of the system.