A Framework Analysis for Managing Explicit Feedback of Visitors of a Web Site

Clemens Schefels
Institute of Computer Science, Goethe University Frankfurt am Main
Robert-Mayer-Straße 10
60325 Frankfurt am Main, Germany
schefels@dbis.cs.uni-frankfurt.de

Roberto V. Zicari
Institute of Computer Science, Goethe University Frankfurt am Main
Robert-Mayer-Straße 10
60325 Frankfurt am Main, Germany
zicari@cs.uni-frankfurt.de

ABSTRACT

In this paper we present a framework analysis for managing the feedback explicitly given by visitors of a Web site. We introduce the concepts of scope, filtering, and relevance profiles for managing users' feedback, and show their applicability by using Gugubarra as a reference system, a prototype developed by DBIS at the Goethe University of Frankfurt, for creating and managing user profiles of Web visitors.

Categories and Subject Descriptors

J.4 [**SOCIAL AND BEHAVIORAL SCIENCES**]: Economics; H.3.4 [**INFORMATION STORAGE AND RETRIEVAL**]: Systems and Software—*User profiles and alert services*

General Terms

Management, Design

Keywords

Web User Profiles, Explicit User Feedback, Gugubarra.

1. INTRODUCTION

An important issue in the management of a Web-based user community, where users are registered to a Web portal, is to identify *patterns of users' interest.* In this context, the users' feedback plays a major role.

In this paper, we present a framework analysis for managing the feedback explicitly given by registered visitors of a Web site. We use as a reference system, Gugubarra [14], a prototype system developed by DBIS at the University of Frankfurt, for analyzing the interests of Web site users.

The rest of the paper is structured as follows: Section 2 recalls the basis concepts of Gugubarra that will be used in the rest of the paper. Section 3 presents the main contribution of this paper, our framework for managing explicit user feedback. Section 4 presents a simple example of use of such framework. Section 5 outlines some problem areas where our approach could be applied. Section 6 presents related work. And finally Section 7 presents the conclusions and outline future work.

iiWAS2010, 8-10 November, 2010, Paris, France.

2. GUGUBARRA AND USER PROFILES

We recall briefly in this section the main concepts of Gugubarra which will be used in the rest of the paper.

Gugubarra [14], [6], is a prototype system developed by DBIS at the Goethe University Frankfurt, with the goal to help the owner/manager of a Web site, to better understand the supposed interests of users registered to his Web site.

Gugubarra generates for each registered user two profiles which collect data related to the user:

An *Obvious Profile*, (OP), which stores the data explicitly given by a user, e.g. name, age, address, e-mail address, etc., for example during the initial registration process and by any updates later on.

A *Non-Obvious Profile*, (NOP), which stores behavioral data not explicitly given by the user, but automatically created by analyzing the user behavior on the Web site.

The behavioral data stored in the NOP indicates, e.g., which pages a user has visited, and which actions he has performed on that Web page. Most of this information is extracted out of the Web server log, but Gugubarra has refined the common click-stream analysis [17], [9], by extending it with new concepts, namely: *zones*, *topics*, *actions*, and *weights* [6], [5]. We describe these concepts briefly here.

A *zone* consists of one or more Web pages, or a set of parts of a Web page. For each zone a list of *topics* is defined, which describe the content of the zone.

Each topic in turn, has an associated *weight.* Topic weights are defined by the owner of the Web site and are used to define the "relative importance" of the given topic with respect to the zone where the topic is defined.

For each zone, a set of *actions* that can occur in that zone are defined. Clicking a link, downloading a file are examples of possible actions. Each action has also an associated *weight*, which indicates the relative importance of the specific action in the context of the zone where the action can occur.

The calculation of a NOP for a user u_m at a given time t_n, with respect to a topic T_i, is automatically computed by Gugubarra using the following formula:

$$NOP_{u_m,t_n}(T_i) = a * ActP(T_i) + b * DurP(T_i) \quad (1)$$

where *ActP* is the so called *Action Profile* and *DurP* is the so called *Duration Profile*.

In the calculation of the NOP, we can decide how important are *actions* (expressed by the Action Profile) and *time duration* (expressed by the Duration Profile), by setting the two parameters a and b in the formula. We can assign values between 0 and 1 to a and b, with the condition that their sum must be 1.

An *Action Profile* takes into account the activities a user has performed on a given zone with respect to a topic, and it is defined with the following formula:

$$ActP(T_i) = \frac{\sum_q (\sum_t aw_t * v(T_i, Z_q))}{\sum_s aw_s} \quad (2)$$

For each zone Z_q with topic T_i, the associated topic weight v is multiplied with the sum of the weight of all actions aw_t that occur in that zone. The result is then normalized by the sum of the weights of all occurred actions. This is in order to normalize the results and obtaining values between 0 (no interest) and 1 (max. interest) for the specific topic T_i.

A *Duration Profile* computes the supposed interests of a user, for a topic T_i taking into account the *time* (*duration*) spent by the user on the page, which contains the topic, and it is calculated as follows:

$$DurP(T_i) = \frac{\sum_j (duration(P_j) * v(T_i, P_j))}{\sum_k duration(P_k)} \quad (3)$$

For a topic T_i, we sum up the weights of the topic T_i in all pages for the zones that contain the topic T_i. The result is then multiply with the time the user has spent on each page. Finally, the result is normalized by dividing it by the total time the user spent on the Web site.

Technically, all NOP-profiles are vectors, and store the *supposed* interest of each user u_m related to a topic T_i at time t_n. Each row contains the calculated interest rate of the user for a topic. The values of the interest rate are between 0 and 1, while 1 indicates high interest and 0 indicates no interest for a topic.

Figure 1 displays an example of a NOP, where we calculated the data for a user based on his behavior, showing a supposed low interest in topic T_1 (0.3), high interest in topic T_2 (1.0), and no interest in topic T_3, (0.0).

$$NOP_{u_m,t_n} = \begin{pmatrix} 0.3 \\ 1.0 \\ 0.0 \end{pmatrix} \begin{matrix} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{matrix}$$

Figure 1: NOP for a user u_m, defined for three topics T_1, T_2, and T_3.

3. A FRAMEWORK FOR MANAGING EXPLICIT USER FEEDBACK

In this section, we consider the problem of how to manage *explicit* user feedback. We also look at the notion of *consistency*, that is, to what extent the behavior of a user (expressed by his NOP) differs from his declaration of interest given with an explicit feedback.

We assume that a user is willing to give a feedback, indicating values associated to a list of topics. This is obviously not the only way for a user to give an explicit feedback, but this method will be used for simplicity in the rest of this paper. We are aware of the limitations of explicit user feedback as indicated in [10], where studies of eBay's reputation system have shown the difficulty to elicit user feedbacks without some sort of incentive.

In the literature (see Section 6, Related work), a user feedback is often used in the field of information retrieval for ranking search results to predict user preferences, but not, as in this paper, in the context of integrating the feedback in the user profile independent from a search.

In the rest, we assume that a user is aware and has given permission that a Non-Obvious Profile (NOP) is generated automatically and is kept for him.

3.1 User Feedback Profile

We assume we ask a user for an explicit feedback (see *Step 3.* below), by asking the user to define his interest with respect to a set of predefined topics, giving a numerical value between 0 and 1 for a each topic: with 0 indicating no interest, and 1 indicating much interest for a specific topic T_i.

To capture this user information, we define a *Feedback Profile* as a vector, similar in structure to a NOP. Figure 2 shows an example of such Feedback Profile, for the same user u_m and for the same three topics, T_1, T_2, and T_3.

$$FP_{u_m,t_n} = \begin{pmatrix} 1.0 \\ 0.5 \\ 0.0 \end{pmatrix} \begin{matrix} \leftarrow T_1 \\ \leftarrow T_2 \\ \leftarrow T_3 \end{matrix}$$

Figure 2: FP for user u_m, defined for three topics T_1, T_2, and T_3.

We can see from the example, that the user has given as a feedback different interest values for T_1, T_2, while confirming the no interest in T_3. We assume in the rest, that we capture the explicit feedback given by a user in his associated Feedback Profile.

Our framework for managing explicit user feedback for a community of registered users of a Web site is composed of several steps:

Step 1: Definition of a Scope

Step 2: Definition of a Filter

Step 3: Obtaining Explicit Users Feedback

Step 4: Filtering the User Feedbacks

Step 5: Clustering

Step 6: Consistency check

Step 7: Interpreting the results of the consistency check

Each step is detailed in the rest of this section.

3.2 Scope and Filter for Users' Feedback

Step 1: Define a scope for the users' feedback.
The meaning of this initial step is to analyze for which users and for which topics (*scope*) we want to take into consideration the users' feedback. With this step we define a *scope*, that is, we set a *cluster of users*, and a *cluster of topics* for which we apply the *filtering* (*Step 2.*) of the user feedback.

A cluster of users can be defined in different ways: e.g. based on their behavioral patterns, and/or based on the user personal data, etc. No matter how the cluster of users is defined, we impose the condition that a cluster of users has at least one user in it. In the extreme case it may contain all registered users.

A cluster of topics can also be defined in different way: e.g. giving a priority list to the list of topics defined for the Web site, and/or taking into account a possible hierarchy or classification of topics, etc. Here we assume for simplicity that a cluster of topics is a subset of the list of all topics defined for a Web site, and we allow that such cluster may contain no topics.

Step 2: Define a filter.
In order to define the *relative* importance of a user feedback, we introduce in this step the notion of a filter, as a function f, which results in a value between 0 and 1. If the result value of f is near to 1, then the user feedback has a substantial influence for the associated cluster of users and/or topics.

While a low value of f, close to 0, decreases the influence of the user feedback. With $f = 0$, the user feedback is *not* taken into account for the associated cluster of users and/or topics.

With f, we decide here how much impact the user feedback should have for a given cluster of users and/or topics. We call this step *filtering*.

f is a function defined for a *scope*, that is a given cluster of users and a given cluster of topics, resulting in an integer between 0 and 1:

$$f(\underbrace{Cluster_{u_i}, Cluster_{T_i}}_{scope\ S_i}) \rightarrow\ between\ 0\ and\ 1 \tag{4}$$

We can apply the same function f with different resulting values to different input scope, i.e. $Cluster_{u_i}$ and $Cluster_{T_j}$. This is a flexible way to handle user feedback.

For example, if the $Cluster_{u_i}$ contains all registered users, then the same filter is set for all users, that is, we treat the feedback in the same way for all users.

But, if the $Cluster_{u_i}$ contains a *subset* of all registered users, this means we apply the specific filter value only for the users who belong to the input cluster. If we want to treat the feedback of users belonging to another cluster in a different way, we simply apply the filter with a different return value to them.

Moreover, if we want to handle in a special way the user feedback related to a *specific* set of topics, we simply group the topics in a cluster and apply a specific filter value. In this way, we can for example decide that the users feedback related to a specific topic T_1 is more important (or less important), than the one given for another topic T_2.

3.3 Obtaining Explicit User Feedback

For *Step 3*, the process of obtaining an explicit user feedback may vary. In this paper, for simplicity we restrict our attention to three cases:

i. The user is presented a set of topics $T_1 ... T_i$, and it is asked to give feedback by indicating a number between 0 (no interest) and 1 (very interested) for each topic in the list. The user feedback is then stored as it is, in a Feedback Profile, FP, associated to the user.

ii. The user is first showed the value of his automatically calculated Non-Obvious Profile for a set of topics $T_1 ... T_i$, and then it is asked to give a feedback indicating a number between 0 (no interest) and 1 (very interested) for each topic in the NOP. The user feedback is then stored as it is, in a separate Feedback Profile, FP, associated to the user.

iii. The user is presented with his last Feedback Profile (FP) for a set of topics $T_1 ... T_i$, and it is asked to give a feedback, indicating a number between 0 (no interest) and 1 (very interested) to each topic in the FP. The user feedback's feedback is then stored as it is, in the Feedback Profile, FP, associated to the user.

It is not in the scope of this paper to discuss the different semantics and implications of these different ways of soliciting users' feedback. We plan to study this research issue in a forthcoming paper.

3.4 Filtering the Users Feedback

With *Step 4*: Applying the scope and filter to the users feedback, we consider the scope and the filter set for it, and we create for each user who has given a feedback and for which we have a NOP, a new profile called *Relevance Profile* (RP). A Relevance Profile is a vector containing values between 0 and 1.

The process of applying the scope and filter to the users feedback works as follows:

At a given time t_n given a scope S_i defined by a cluster of users $Cluster_{u_i}$, and a cluster of topics $Cluster_{T_i}$, and a filter f_i for it. For each user u_m belonging to the scope S_i, and given a set of topics T_i, also belonging to the scope S_i, we create a *Relevance Profile* for the user u_m, related to the topics T_i, using the following formula:

$$RP_{u_m,t_n}(T_i) = \frac{NOP_{u_m,t_n}(T_i) + f_i(S_i) * FP_{u_m,t_n}(T_i)}{a + b + f_i(S_i)} \tag{5}$$

RP is calculated by integrating the two available profiles for the user: the Non-Obvious Profile (NOP), which is automatically calculated, based upon the behavior of the user (i.e. his Action Profile and his Duration Profile), with the explicit feedback given by the user indicated in his Feedback Profile (FP), and filtered f_i to the relevant scope S_i.

In the formula, RP sums up the NOP applying the filter to FP, and it is normalized to obtain values between 0 and 1, which corresponds to the domain of the NOP.

After this step, we have for each registered users who has given a feedback, besides a NOP and FP also a Relevance Profile (RP). The benefits of RP is that it integrates in a flexible way into one single user profile both calculated data

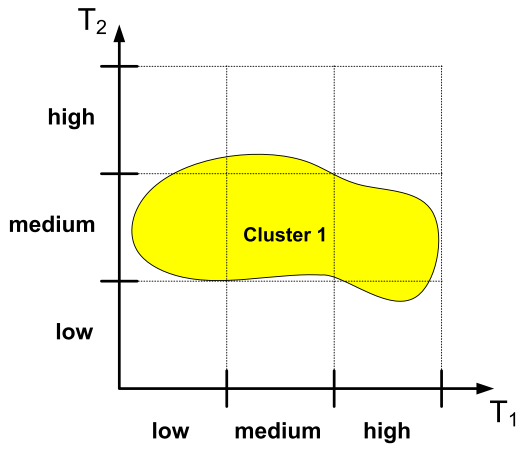

Figure 3: Cluster of RPs without taking into account users' feedback (filter $f = 0.0$).

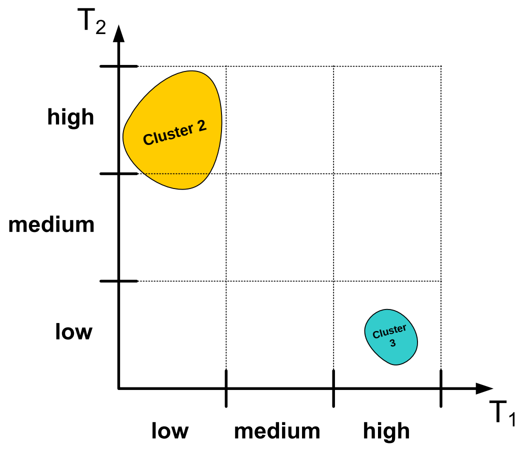

Figure 4: Cluster of FP.

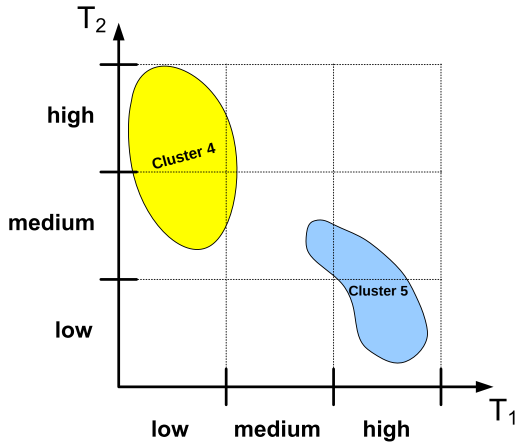

Figure 5: Cluster of RP with $f = 1.0$.

and feedback of the user. In Section 5 we will show a few area of applicability for the RPs.

For very high number of users, it is better to work with clusters of Relevance Profiles rather than RPs for individual users, as it is illustrated in the next section.

3.5 Clustering

When the number of registered users is very high (e.g. millions of users), it is better to work with *clusters of Relevance Profiles*. In *Step 5*, RPs are therefore clustered. RPs can be clustered together in several different ways, for example grouping together users with RPs showing similar behavioral patterns of interest. We recall that a RP integrates the behavioral data calculated by the NOP, and the filtering of the user feedback stored in the Feedback Profile.

We show the applicability of clustered RPs with an example. Assume we have a large community of registered users for a Web site with two topics. We use a fuzzy clustering algorithm [4], [11] for clustering the users RPs, and we assume that all RPs are generated with a filter $f = 0.0$ (meaning we did non take into consideration the user feedback (if any) in the calculation of the RPs). In the example, we assume the scope S_i consists of a $Cluster_{u_i}$ including *all* users and a $Cluster_{T_i}$ including two topics T_1 and T_2. We obtain the cluster for RPs depicted in Figure 3.

The size of the cluster clouds (*Cluster 1*), describes the number of users. The X-axis displays the interest values for topic T_1 and the Y-axis the interest values for topic T_2. The resulted centroids of the fuzzy clustering algorithm are shown in Table 1 below. The centroids are representative of pattern of user's interest.

Since we have used a filter $f = 0.0$, *Cluster 1* represents RPs taking into account only the calculated behavioral data stored in the NOP of the users, and not the users feedback (if any). In the example *Cluster 1* is a one big cluster. By looking at this cluster of RPs, we can only infer that all users who seem to have a medium interested in topic T_1 have also a medium interest in topic T_2.

At this point, we may ask the users for an explicit feedback asking them to indicate their interest in T_1 and T_2. A user feedback is stored in a Feedback Profile for each user. For large number of users, we can also cluster the Feedback Profiles (FPs). In the example, we cluster the FPs by using the same algorithm we have used for clustering RPs. We obtain the two clusters of FPs displayed in Figure 4: from the cluster of users' feedback, it results that users with high interest in T_1, have only low interest in T_2 and vice versa. Table 2 shows the corresponding centroids.

Now we take into account the user feedback, and we recalculate the RPs this time, by setting a filter different than 0, in the example we use $f = 1.0$ (i.e. we take in full consideration the users feedback when generating RPs).
We then cluster the generated RPs, and this results in the two clusters of RPs in Figure 5.

The proper centroids are summarized in Table 3. Having taken into account this time both the NOP and the FP, we can now interpret the results: visitors who seem to be interested in T_1, show only low interest in T_2, while users who seem to be interested in T_2, show a low or medium interested in T_1.

Our approach is flexible as we can generate RPs by scope, using different filter f values.

Table 1: Centroids for Cluster 1.

	Cluster 1		caption	
	T_1	T_2	■	= most users
High Interest	▦	⊠	▦	= some users
Medium Interest	■	■	⊠	= few users
Low Interest	▦	⊠	□	= no users

Table 2: Centroids for Cluster 2 and 3.

	Cluster 2		Cluster 3		caption	
	T_1	T_2	T_1	T_2	■	= most users
High Interest	□	■	■	□	▦	= some users
Medium Interest	□	⊠	□	□	⊠	= few users
Low Interest	■	□	□	■	□	= no users

3.6 Consistency Check and Interpretation of the Results

In *Steps 6* and *7*, we turn now our attention to the notion of user *consistency*. That is, we look if the behavior of a user (calculated and expressed by his NOP) is *consistent* with his declaration of interest, expressed in his explicit feedback (captured in the FP), and filtered and generated in his RP.

We do not intend to judge the intention of the users, but rather measure the possible *discrepancy* between actions performed by a user and his declaration of interest expressed by the feedback. We will show in Section 5 some possible domain of applicability of our approach for measuring user *consistency*.

In order to have a measurable indication of user consistency, we compare for a given user, his calculated RP with respect to his NOP and his FP.

When FP is not equal to the NOP, and $f \neq 0$, we have that NOP, FP, and RP do represent different values. We can calculate the difference between RP and NOP, and between RP and FP.

Comparing RP with NOP: Given a NOP and a RP of a user, and let f be a filter value, there are three possibilities:

i. *Confirming effect*, denoted with "=". If all values of RP equal all values of the NOP, then we say that the RP *confirms* the NOP.

ii. *Positive adjusting effect*: denoted with "+". If some values of RP are higher than some values of NOP, then the RP *positive corrects* the calculated values of the NOP.

iii. *Negative adjusting effect*, denoted with "-". If some values of RP are smaller than some values of NOP, the RP *negative corrects* the calculated values of the NOP.

The above definition does not apply if for the same RPs, some values are higher and some are lower with respect to the NOP.

Comparing RP with FP: Given a FP and a RP of a user, and let f be a filter value, there are three possibilities:

i. *Confirming effect*, denoted with "=". If all values of RP equal all values of FP, then the RP *confirms* the FP.

ii. *Positive adjusting effect*: denoted with "+". If some values of RP are higher than the corresponding values of FP, then RP *positive corrects* the FP.

iii. *Negative adjusting effect*, denoted with "-". If some values of RP are smaller than the corresponding values of FP, then RP *negative corrects* the FP.

The above definition does not apply if for the same RPs, some values are higher and some are lower with respect to the FP.

These measured notions of user consistency are dependent on the value of f we use, as it will be illustrated in the next section with a simply example. The higher the value of f, the more influence a FP has on the calculation of a RP (indicated with "++" or "- -").

For large amount of users, the same measure can be made at cluster levels, that is we can compare cluster of RPs with cluster of FPs and clusters of NOPs.

4. AN EXAMPLE OF CALCULATING RELEVANCE PROFILES

To illustrate the calculation and interpretation of the RP, and the influence of f, we show a simple example using Gugubarra (Figure 6):

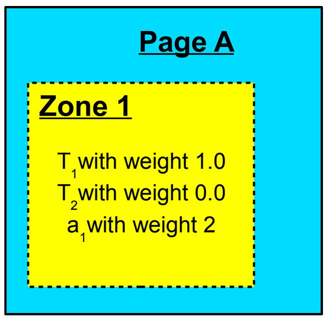

Figure 6: Page with one zone, two topics, and one action.

We assume to have a Web page A, with one zone 1 with two topics T_1 and T_2.

Topic T_1 has weight 1.0, while the weight of T_2 is 0.0. This means that for the owner of the Web site T_1 is very important in zone 1, while T_2 is not important. One action a_1 is defined in zone 1 with weight 2.

We assume two acting users u_1 and u_2, with similar behavior, both visiting page A for five minutes each and both

Table 3: Centroids of Cluster 4 and 5.

	Cluster 4		Cluster 5		caption	
	T_1	T_2	T_1	T_2	■	= most users
High Interest	□	■	■	□	▩	= some users
Medium Interest	⊠	▩	⊠	▩	⊠	= few users
Low Interest	■	□	□	■	□	= no users

performing the same action a_1.

Assume the NOPs are calculated with $a = b = 0.5$ for both users, that is ActP and DurP have the same relevance. As result both users have the same NOP because they behave exactly in the same way, both seem to be very interested in T_1 and show no interest in T_2:

$$NOP_{u_1}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = 0.5 * \overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{ActP} + 0.5 * \overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{DurP} = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

$$NOP_{u_2}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = 0.5 * \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} + 0.5 * \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

We ask now the two users for a feedback: what is their interest for the topics T_1 and T_2? Assume both users give their feedback for topic T_1 and T_2 indicating the following values captured in their Feedback Profiles:

$$FP_{u_1}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}, \; FP_{u_2}\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}$$

From the feedback, user u_1 confirms the value calculated in his NOP, but user u_2 gives different values than in his NOP. We now calculate the two RPs, assuming first a filter $f = 1.0$ (we assume both users and topics to belong to the same scope). The RP for each user is calculated as follows:

$$RP_{u_1}(T_1) = \frac{\overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{NOP} + 1 * \overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{FP}}{1+1} = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

$$RP_{u_2}(T_1) = \frac{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} + 1 * \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}}{1+1} = \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix}$$

If we use a different filter, say $f = 0.1$ this means we have a low consideration of the users' feedback, when generating the RPs, we obtain following result:

$$RP_{u_1}(T_1) = \frac{\overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{NOP} + 0.1 * \overbrace{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}}^{FP}}{1+1} = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}$$

$$RP_{u_2}(T_1) = \frac{\begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix} + 0.1 * \begin{pmatrix} 0.0 \\ 1.0 \end{pmatrix}}{1+1} = \begin{pmatrix} 0.9 \\ 0.1 \end{pmatrix}$$

Table 4, sums up the results of the calculations above using $f = 1.0$ and $f = 0.1$, showing the measured user consistency.

In the table the NOP calculated for user u_1 is equal to the feedback. Therefore the RPs, no matter which value f has, and NOPs have identical values for both topics T_1 and T_2. This is what we called *confirming effect*. The same effect is also seen if f is set to a value of 0.0–than the feedback has no effect on the RP calculation and therefore the RP equals always the NOP. The confirming effect is marked as equal sign "=" in Table 4.

Different is the situation for u_2. For u_2 from the NOP the user seems to be interested in topic T_1, while the feedback says that he is not interested in T_1 at all. We do not know which of the values reflects the real interest of user u_2. So the Web site owner has to decide whether he trusts more the explicit feedback (FP) or more the implicit feedback (NOP) of the user. He can set the filter for the FP accordingly in the RP. In Table 4 the RP with $f = 1.0$ reflects for example a high trust in the users explicit feedback, while the RP with $f = 0.1$ indicates for example low trust. For topic T_1 the *negative adjusting effect* of the RP is shown: the values of the NOP are decreased because of the FP. The effect is more distinctive the higher the value of f is set. Even the difference between FP and NOP influences the effect.

A positive adjusting effect is to shown on topic T_2, where the FP increases the NOP values. In Table 4, a single plus sign "+" symbolizes a "weak" positive adjusting effect, a minus sign "-" a "weak" negative adjusting effect. If the effects are more "significant" two plus/minus signs "++"/"- -" are used. "++"/"- -" means the calculated values of RP are closer to FP than before the feedback was given.

This example shows that our measures of user consistency depends on the value of f we use. This is consistent with our approach of having a flexible way to decide to which extend the users feedback are taken into consideration, when calculating the supposed interest of the users. The setting of an appropriate filter of a given scope allows that.

In [5] we presented a first user test based evaluation of the capability of our system.

Much work needs to be done in this area, in Section 7 we outline some of our planned future work.

5. APPLICABILITY OF RELEVANCE PROFILES

The flexible way we manage the user feedback by incorporating it, filtered together with the NOP data, into a single coherent profile per user, the Relevance Profile, make it attractive for a variety of practical applications. Here we list some of them.

RPs are well suited to *cluster* and *compare* users, with no need to consider several different profiles for the same user.

RPs make it easy to add new form of user feedbacks (e.g. *eye tracking*) by integrating them into the RPs, and setting

Table 4: For users u_1 and u_2: FP, NOP, generated RPs and different users' consistency.

User	Topic	FP	NOP	RP (*f=0.1*)	Effect	RP (*f=1.0*)	Effect
u_1	T_1	1.0	1.0	1.0	=	1.0	=
u_1	T_2	0.0	0.0	0.0	=	0.0	=
u_2	T_1	0.0	1.0	0.9	-	0.5	- -
u_2	T_2	1.0	0.0	0.1	+	0.5	++

a filter defined reusing different available analysis methods.

RPs are useful to solve the so called "*cold-start problem*" defined by Maltz and Ehrlich [13]: if a new user registers to the Web site, at the beginning we have no information about his interests since the user did not do any action yet. With the integration of the user Feedback Profile into his RP, we can solve this problem by asking the user about his interests. If the user is willing to give a feedback at the time of the registration, we integrating it into his RP.

RP are useful in cases of *missing information*. Sometimes there is missing information about a topic in the user profile, for example because the user never visited a page with this specific topic. When asked for a feedback, if the user feedback indicates a high interest in this topic, it is possible that the user did not find the topic on the Web site but he is interested in. A possible reason could be that the Web site has a bad design or bad navigation, so that the user was not able to find the available Web pages with this topic. If the feedback indicates instead a low interest in these topics, the user is simply not interested and did therefore not visit the Web pages with this topic. So the missing interest information is not caused by a bad designed Web site. By integrating this feedback into his RP allows us to evaluate how effective the Web site is.

Feedback error compensation is another problem domain where RPs are useful. Assume a user misunderstood a feedback questionnaire–for example giving wrong answer for a specific topic T_x: e.g. suppose the user is not interested in T_x, but gave a feedback "very interested in T_x" because of this misunderstanding. In the RP we compensate this misunderstanding, by taking into account the NOP data which shows that the user did not visit any Web page (or did any actions related to) with topic T_x. So, in the generated RP for this user, we correct this and the resulting interest for topic T_x for the user are set to a lower value (how much lower the value depends on the setting of f).

6. RELATED WORK

User feedback is often used in the field of information retrieval, for ranking search results to predict user preferences. White et al. [18] examine the extent to which implicit feedback can act as a substitute for explicit feedback, where searchers explicitly mark documents as relevant. With our approach we combine NOP and FP in one profile and use this data for analysis.

Agichtein et al. [1] examine alternatives for incorporating feedback into the ranking process and explore the contributions of user feedback compared to other common Web search features. The implicit feedback caused significant improvements on the quality of Web search result rankings. With the NOP we integrate implicit feedback into the RP to improve its expressiveness.

Lin et al. [12], define a so called *reputation manager* which collects feedback ratings from its clients after each transaction. In our approach Web site owners can easily weight the feedback of users with respect to their reputation. For example using a different filter f for expert and non-expert users or with a specific consistency history, we can get different weights reflecting their reputation levels.

In [15] algorithms for learning and revising user profiles are defined that can determine which World Wide Web sites on a given topic would be interesting to a user. The authors use a Bayesian classifier for this task. In [8] the authors examine the reliability of implicit feedback generated from click through data in WWW search. This is done by analyzing the users' decision process using eye tracking and comparing implicit feedback against manual relevance judgments, and concluded that clicks are informative but biased.

Zigoris [19] uses a Bayesian adaptive user profiling with explicit and implicit feedback, and address the cold-start problem, proposing that implicit feedback should be combined with explicit feedback to get a stable base for prediction. This is related to our future work where we want to use the RP for predicting user interest changes.

Fink, Kobsa, and Schreck [3] define a public accessible, personalized hypermedia system with an adaptive user interface and content. They also distinguish between obvious and non-obvious user profile information. In addition, to overcome the cold-start problem, they use so called 'stereotypes'.

7. CONCLUSIONS AND FUTURE WORK

The main result of this paper is the definition of a framework analysis for managing explicit user feedback, which integrates user behavior data automatically computed and user feedback, filtered into a Relevance Profile (RP). The calculation of RPs, their interpretation, and applicability was presented with the help of some examples.

In the future, we want to analyze different factors influencing the feedback of a user, e.g. which kind of questionnaire forms [16] should be used, how often should we ask for feedback, what new Web 2.0/Web 3.0 techniques, e.g. tagging [7], can support feedback collection. Moreover, we want to use the RPs as a basis for *predicting changes in user interest*.

We also intend to conduct some empirical experiments using our framework in order to validate our results, and to measure how the combination of framing strategies and user feedback influence the user's choice of content on the Web [2].

8. ACKNOWLEDGMENTS

We would like to thank Natascha Hoebel, Karsten Tolle, and Naveed Mushtaq of the Gugubarra team, for their valuable support and fruitful discussions.

9. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] I. Constantiou, N. Hoebel, and R. V. Zicari. Reputation, framing strategies and user's choice of content on the web: an empirical study. *Concurrency and Computation: Practice and Experience*, 22(7):872–889, 2010.

[3] J. Fink, A. Kobsa, and J. Schreck. Personalized hypermedia information provision through adaptive and adaptable system features: User modelling, privacy and security issues. In *IS&N '97: Proceedings of the Fourth International Conference on Intelligence and Services in Networks*, pages 459–467, London, UK, 1997. Springer-Verlag.

[4] N. Hoebel and S. Kreuzer. Cord: A hybrid approach for efficient clustering of ordinal data using fuzzy logic and self-organizing maps. In *6th International Conference on Web Information Systems and Technologies (WEBIST 2010)*, Berlin, Germany, 2010. Springer.

[5] N. Hoebel, N. Mushtaq, C. Schefels, K. Tolle, and R. V. Zicari. Introducing zones to a web site: A test based evaluation on semantics, content, and business goals. In *CEC '09: Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, pages 265–272, Washington, DC, USA, 2009. IEEE Computer Society.

[6] N. Hoebel and R. V. Zicari. Creating user profiles of web visitors using zones, weights and actions. In *Tenth IEEE Conference On E-Commerce Technology (CEC 2008) And The Fifth Enterprise Computing, E-Commerce And E-Services (EEE 2008)*, pages 190–197, Los Alamitos, USA, 2008. IEEE Computer Society Press.

[7] M. Jazayeri. Some trends in web application development. In *FOSE '07: 2007 Future of Software Engineering*, pages 199–213, Washington, DC, USA, 2007. IEEE Computer Society.

[8] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[9] S. Jung, J. L. Herlocker, and J. Webster. Click data as implicit relevance feedback in web search. *Inf. Process. Manage.*, 43(3):791–807, 2007.

[10] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 1026–1027, New York, NY, USA, 2003. ACM.

[11] D.-W. Kim, K. H. Lee, and D. Lee. Fuzzy clustering of categorical data using fuzzy centroids. *Pattern Recogn. Lett.*, 25(11):1263–1271, 2004.

[12] K.-J. Lin, H. Lu, T. Yu, and C.-e. Tai. A reputation and trust management broker framework for web applications. In *EEE '05: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05) on e-Technology, e-Commerce and e-Service*, pages 262–269, Washington, DC, USA, 2005. IEEE Computer Society.

[13] D. Maltz and K. Ehrlich. Pointing the way: active collaborative filtering. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 202–209, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[14] N. Mushtaq, P. Werner, K. Tolle, and R. Zicari. Building and evaluating non-obvious user profiles for visitors of web sites. In *IEEE Conference on E-Commerce Technology (CEC 2004)*, pages 9–15, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[15] M. Pazzani and D. Billsus. Learning and revising user profiles: The identification ofinteresting web sites. *Mach. Learn.*, 27(3):313–331, 1997.

[16] P. Van Schaik and J. Ling. Design parameters of rating scales for web sites. *ACM Trans. Comput.-Hum. Interact.*, 14(1):4, 2007.

[17] B. Weischedel and E. K. R. E. Huizingh. Website optimization with web metrics: a case study. In *ICEC '06: Proceedings of the 8th international conference on Electronic commerce*, pages 463–470, New York, NY, USA, 2006. ACM.

[18] R. W. White, J. M. Jose, and I. Ruthven. Comparing explicit and implicit feedback techniques for web retrieval: Trec-10 interactive track report. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, 2001.

[19] P. Zigoris and Y. Zhang. Bayesian adaptive user profiling with explicit & implicit feedback. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 397–404, New York, NY, USA, 2006. ACM.