A Hybrid Approach for User Profiling

Danny POO, Brian CHNG and Jie-Mein GOH
Department of Information Systems
School of Computing
National University of Singapore
Lower Kent Ridge Road, Singapore 119260
{dpoo, chngteck, gohjm}@comp.nus.edu.sg

Abstract

With the growing amount of information being digitized, users find it difficult to obtain the most relevant information that is hidden in the deluge of information returned to them by search engines. In this paper, we describe a conceptual model using a hybrid approach based on user profiling approaches and information filtering techniques that can alleviate the problem of information overload. We discuss the concepts, namely static content profiling, static collaborative profiling, dynamic content profiling and dynamic collaborative profiling, followed by the design and implementation of a library search facility, which has employed these concepts. Preliminary experiments conducted have shown that users have benefited from our prototype system. It appears that there is value in employing these new concepts that we have proposed in the design and implementation of future information retrieval system for better retrieval.

1. Introduction

Search engines, meta-search engines and other search facilities have been developed to help users look for information that users find useful and return a list of links to resources that have matched specific search criteria. Despite efficient information retrieval technologies, users are still not satisfied with the search process and the results presented. Studies have shown that users enter very few query terms [3] especially when they are not familiar with the topic. Hence, users are often inundated with a large amount of links returned due to the generality of the terms used during a search, and the lack of specific information on the user who utilizes these facilities.

With the rapidly growing amount of information, especially on the web, users are often overwhelmed by the large amount of information they have to go through and experience the problem of information overload. Information overload is a situation whereby the individual is no longer able to effectively process the amount of information he or she is exposed to. The perceived utility of the information presented to the users decline can result in a lower decision quality in a given time. This can be alleviated by the provision of more user information at the time when the search was performed.

In our study, we propose a hybrid approach of combining concepts used in user profiling and information filtering and attempt to resolve the problem of information overload and to improve the precision of current search facilities. The notion of this conceptual model is for the complete consideration of user profiling in the design and implementation of future information retrieval systems. Our proposed technique consists of four main concepts, namely, static content profiling, dynamic content profiling, static collaborative profiling and dynamic collaborative profiling. These concepts have been derived from a combination of user profiling and information filtering concepts. From our preliminary experiments, we found that this new approach appears to have improved the precision of a search facility and reduced the amount of irrelevant information returned to the user.

The rest of this paper is structured as follows. We will first discuss related work and background concepts that we have employed. Following this, we present and explain our proposed concepts. The third section will detail how the new concepts have been applied on a search facility of a library system. The fourth section will touch on the implementation details of the system, SNAS, that we have built based on these new concepts. Finally, we present our experimental results and give a conclusion.

2. Background & Related Work

Several approaches in information retrieval have been developed to produce better search results or to guide

0-7695-1874-5/03 $17.00 © 2002 IEEE

users towards more relevant results. In INVAID [6], the system receives explicit user feedback through ratings of relevant pages and suggest pages of interest to users based on the feedback of the user coupled with filtering strategies. Pazzani et al [10] and Aniscar and Tasso [1] created intelligent agents that will analyse user feedback based on ratings defined by the user on the visited page as a measure of user interest. They perform an extended navigation of related pages and graphically show the set of the pages found, classified according to the user's interest. These systems request users to provide explicit feedback on documents in terms of ratings or preferences. Employing user's feedback to improve systems had shown to be effective. However, in the real world, it is difficult to ensure that all users will voluntarily offer their feedback to such systems. The approach taken here will not only focus on the filtering techniques, but the means to get a user's feedback implicitly.

There has been previous literature discussing various approaches to resolve the problem of sieving out relevant information from a larger set of information namely through user profiling and information filtering techniques. Reviewing previous studies on user profiling have led us to identify two main types of profiling approaches namely, static profiling and dynamic profiling:

Static profiling is the process of analysing a user's static and predictable characteristics. Such information usually comes from users themselves e.g. electronic registration or survey forms. Through static profiling we usually know what kind of information the user is generally interested as soon as the user have supplied the information. There are several problems when we rely solely on static profiling i.e., getting user's profile through manual input. Firstly, the profile is static, and is only valid for a certain period of time until the user changes their interest. Hence, a static profile degrades in quality over time. In addition, the input is based on the individual's interest, prone to users' subjectivity and may not accurately reflect an objective view that can infer the interests of other users with similar interests.

Dynamic profiling on the other hand is the process of analysing a user's activities or actions to determine what the user is interested in. In this aspect of profiling, the user's behaviour is of interest to us and it is sometimes referred to as behavioural profiling. Although the users' information needs are captured at real time, the more general interests of the user cannot be traced.

Similarly, there has been a lot of literature that discussed on information filtering techniques. Two kinds of approaches for information filtering have been presented in previous literature:

Content based filtering compares the contents of items associated with a user profile and selects those documents whose contents best match the contents of another user profile using some similarity measures. In Avery's work [2], a system that receives explicit user feedback through ratings of relevant pages uses filtering strategies to suggest pages of interest to users was developed. Stewart et al [13] and Oard [9] created intelligent agents that analyse the user feedback based on well defined ratings of visited pages as a measure of user interest. All the above systems request users to provide explicit feedback on documents in terms of ratings or preferences. The content of the profile dominates in this approach and depends on how well the profiles match that of other users. This can be a problem as some users are reluctant to furnish details about themselves. In the real world, it is difficult to ensure that all users will voluntarily offer their feedback to such systems due to the cost of examining and rating an item [12]. Unless the user perceives that there is additional value in participating in such evaluation, the system with all the best filtering strategies may still result in the lack of any ratings at all [5]. Thus, implicit rating is needed such that it removes the cost of examination of an item from any evaluator. In addition, the computational cost of such implicit ratings must be at best hidden away from the user.

Collaborative filtering organizes users with similar interest into peer groups, thus enabling the recommendation of documents considered interesting by peers to other members of that group. Usually this is done by clustering the profiles of different users. Examples are BIRCH [15] and DBSCAN [4]. As this approach relies heavily on user clusters, its effectiveness highly depends on how well the clustering of profiles correlates the users.

Thus, to provide a complete user profiling system, we need to consider the concepts from user profiling and information filtering. In our approach, we will adopt the concepts of user profiling and information filtering by providing explicit and implicit ratings, as well as both content and collaborative filtering to implement a user profiling system.

3. Hybrid Approach to User Profiling

Based on the concepts discussed in previous works, we have defined four new concepts (Figure 1) that will help us identify and categorise our profiling strategies, they are:

- Static Content Profiling
- Dynamic Content Profiling
- Static Collaborative Profiling
- Dynamic Collaborative Profiling

Each individual category is important in creating a profiling system since the strategy used in each category is unique and will contribute information in creating and shaping a user's profile in various ways. Search facilities or search engines, which filter search results and other results generated from user queries can apply these concepts in their system design in order to derive information gathered from these strategies to profile users. We have applied these new concepts on a library search facility to improve on the existing system's retrieval facility. We will describe the strategies to implement each of these concepts in the next section.

	CONTENT	COLLABORATIVE
DYNAMIC	***Dynamic Content Profiling*** refers to gathering of information based on the dynamic changes in the behaviour of the user and filtering only those that represent the user's profile.	***Dynamic Collaborative Profiling*** refers to organising users with similar behaviour into peer groups based on the user's profile and filtering information pertaining to group's interest.
STATIC	***Static Content Profiling*** refers to the gathering of static information regarding the user only.	***Static Collaborative Profiling*** refers to explicitly organising users with similar behaviour into peer groups through user explicit request.

Figure 1. Categorisation of profiling strategies

3.1. Static Content Profiling

Static content profiling refers to the gathering of static information regarding the user usually upon registration. Typically, systems allow users to enter a simple profile when they first register with the system. It is static as the registration is done only once. For instance, in our prototype system, users enter information via the registration interface that allows the system to capture and store their interests. Thus, in any system, user registration is one mode to capture the static content profiling concept.

In our prototype, users are required to register with the system so as to capture the static content profiling information. The user will have to fill in vital information that will help the system to learn more about the user. A model called the W3 model (Figure 2) has been used in designing a comprehensive domain specific user profile for static content profiling.

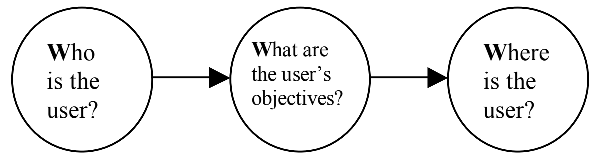


Figure 2. W3 Model

This model comprises of three questions that we used to determine the information that needs to be captured in the user profile. In each step, a set of features $W_i = \{f_1, f_2, \ldots, f_n\}$ are listed. The final user registration, R will consists of all the features derived from each step, $R= W_1UW_2UW_3$. Users will then fill in the necessary information that we have derived from this model. Each profile, P is represented by a set of feature-value pairs i.e. {(fi,vi)} where fi is the feature or keyword and value is the weight assigned to the keyword based on users' profile and feedback. The profile begins as an empty set but subsequent interactions of the user with the system will add or remove new feature-value pairs and whose weights will adaptively change. Capturing static content information is inadequate as it is difficult to capture explicit feedback consistently and is limited by the individual user's knowledge. Therefore, further concepts will be described in the following sections, which are required to address these issues.

3.2. Dynamic Content Profiling

For dynamic content profiling, the system gathers information based on the dynamic changes in the behavior of the user. This means that the system should keep track of the user's behavior during the search process. There are three ways to capture this information:

Monitor User's Actions. Browsing patterns and clicking activity in the interface provide another source of information about users. Such activity is analyzed to determine topics and concepts of interest through off-line data mining. For instance, in our prototype, data that detect the user's reading behavior like which documents the user has borrowed or reserved are recorded so that the system will capture the user's reading habits. The information about the user's reading habits can then be used to recommend items of interest based on the user's past behavior to the user. It is also an implicit way of gathering feedback about a user's preference without having the user to offer explicit views about his/her interest.

Monitor User Search History. Most users will not remember the whole process of their search and how they arrived at the results they wanted but rely on keywords or nodes to help them recollect their search routines.

Whenever a user searches for results the set of keywords used could range from as many as 5 to 20. Thus, it is difficult to remember all the keywords that were used in previous searches, hence, we introduced a data structure that stores all past keyword searches and link each word to one another based on the search text. For example, a user in his/her last search may have used additional keywords recommended by the system to refine the search. However, due to the limited information processing capability of humans, the user may have only remembered the keywords that he/she used in the search and not the keywords that were recommended by the system. The data structure would then be used to help users implicitly recollect keywords that they have used previously.

Monitor User Preference. It is always essential to note what a user is interested in any profiling system. However, this is a difficult task as it is always very difficult to gather feedback without having the user to explicitly indicate his/her preferences. This is especially so if a user is searching over the Internet. However, in our prototype system, our implicit feedback can come in the form of items that have been read. By analysing the documents that have been read during the search, we can determine what sort of items a user is interested as well as determine items a user is less interested. Based on this information, we suggest additional words to augment the query so that we can filter away disinterested records and return only interested ones. The way to do this is to extract the subject keywords that describe a particular item of interest. In our prototype system, this can be done easily by locating the subject in a library classification scheme, associated with the particular item of interest. With the subject description we build a list of interested and disinterested keywords based on the user's feedback like whether a user profile contains domain categories having ratings to show if a category is interesting or disinteresting. We have obtained the list of keywords that are plausibly interesting to the user through the dynamic content profiling approach. Now, the user can use explicit feedback to refine his search. When the user indicates that he is interested in the terms suggested to the user and the term did not exist in the user profile, the term will be added to the interested list of the user profile and the weight of that term is increased. Figure 3 shows the detailed steps.

3.3. Static Collaborative Profiling

This concept refers to explicitly clustering users with similar behaviour through user's explicit request. Every time a new user is added into the system, the system will take a period of time to collect information about the user and to construct the user's profile with information that will aid the system in serving the user's needs. We can reduce the learning curve of the system by reusing a current user's profile by matching the new user's profile with other current user's profile. The categories or terms listed in the user's profile are matched across other users' profiles. If the term or keyword in the user's profile is found in another user's profile, the similarity measure for these two users is increased accordingly. An outline of the profile matching algorithm is given in Figure 4.

User Cluster Assignment. In this technique we group users according to their reading patterns or preferences and this is based on the user's explicit feedback that he/she wants to be placed under this cluster and be informed of items that exist under this cluster. This also means that the loans and reservation patterns shown by the user does not only help the system in knowing what a user wants, but as well as allow individual user's with similar behavior to share information with one another via their recommendations and preferences on items of interest. The clustering algorithm (Figure 5) has been used in our prototype.

```
new_keyword_list  //new set of keywords retrieved via search engine search results
interested_keyword
for each list (interested, indifferent and disinterested)
  if interested_keyword found in interested list
    then increase importance of keyword in the interested list
  end if
  if interested_keyword found in indifferent list
    then remove from indifferent list and place it in interested list
  end if
  if interested_keyword found in disinterested list
    then remove from disinterested list and place it in interested list
  end if
end for
if interested_keyword not found in any of the 3 list
  then place it in interested list and remove from new_keyword_list
end if
for each keyword k in new_keyword_list
  if k found in interested list
    then increase importance of keyword k in the interested list
         and remove from new_keyword_list
  end if
end for
for each remaining keyword k in new_keyword_list
  pattern match against interested_keyword
  if success
    then place it in interested list and remove from new_keyword_list
  end if
  if found in indifferent list
    then remove from indifferent list
  else if found in disinterested list
    then remove from disinterested list
  end if
end for
for each keyword k in indifferent list
  k.weight
  if k.weight <= 0
    remove k from indifferent list and place it in disinterested list
  end if
end for
```

Figure 3. Algorithm For Managing Keyword Preference List

```
Profile best_match
double best_weight
for each existing profile found
  for each attribute type in new profile
    for each same attribute type in existing profile
      for each keyword in existing profile attribute k
        if k is found in existing profile attribute
          weight =+1/total number of keywords in new profile attribute
      End for
    End for
  End for
  average_weight = weight/number of attributes used for comparison
  If best_weight <= weight
    best_match = existing_ profile
End for
```

Figure 4. Profile Matching Algorithm

```
for each user
  Retrieve user loan and reservation data and store in list new_items
  for each item in list new_items
    Check if item is in existing cluster
    if yes
      Update item's rating in cluster
      if user not assigned to cluster
        Assign user to that cluster
        Number of users assigned to cluster + 1
      End if
      Indicate cluster has been modified
      Decrease all other item's rating in cluster
      Remove item from new_items
    End if
  End for
for each item in list new_items
    Check if item can be placed in existing clusters
    if yes
      Add item to cluster
      if user not assigned to cluster
        Assign user to that cluster
        Number of users assigned to cluster + 1
      End if
      Indicate cluster has been modified
      Remove item from new_items
    End if
    If no
      Create new cluster base on subject description and author and keyword
      associated with item
      Remove item from new_items
    End if
End for
```

Figure 5. Clustering Algorithm

3.4. Dynamic Collaborative Profiling

Dynamic collaborative profiling refers to clustering users with similar behavior into peer groups based on the user's profile and filtering information pertaining to group's interest.

System Cluster Assignment. This technique is similar to that in "User Cluster Assignment", but the difference is that the system will cluster users based on dynamic feedback via their loans and reservation patterns.

4. SNAS

In this section, we apply the concepts that we have discussed earlier in a library setting. We will present the architecture and performance of SNAS (Sensitive New Age Search) to demonstrate the feasibility of implementing the hybrid model for improving search facilities. SNAS is a system designed to improve library user's experience when searching by improving the precision of the search and to reduce information overload in a library system. This main goal of the system is to assist users who are searching for specific information and are not able to express their query adequately to retrieve information that are perceived to have high value to the user. SNAS uses the four concepts that we have presented: static content profiling, dynamic content profiling, static collaborative profiling and dynamic collaborative profiling.

4.1. Architecture

SNAS is built using Java Servlet technology and is driven by Apache Web server and tomcat 3.2 a servlet engine. There are two basic components that make up SNAS i.e. the search engine and the backend profiling system (Figure 6). Both of these major components are also sub-divided into smaller ones that work hand in hand to profile users using SNAS.

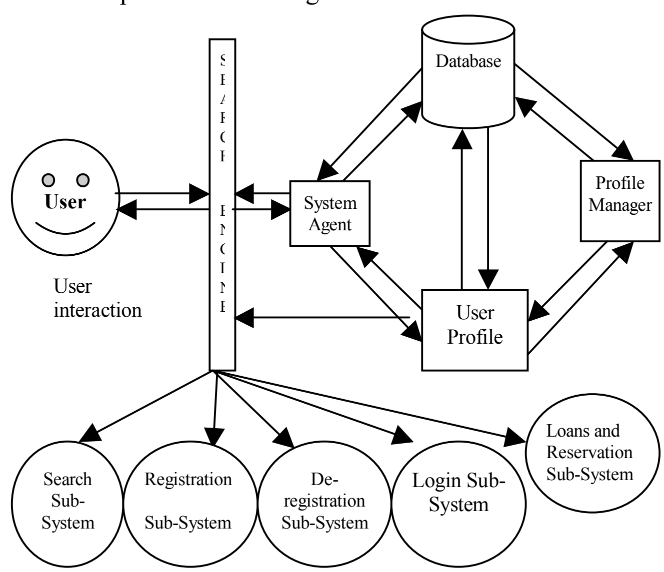


Figure 6. SNAS Architecture

Search Engine. We have utilised an existing search engine, E-Referencer 2.0 [11] that connects to library server's database via the Z39.50 protocol and does relevant ranking on documents or records returned by the Z39.50 server. Besides allowing the user to search for records the search engine is also the primary source of user feedback for the system. The new concepts have been applied and we have identified 5 sub-systems that need to be extended from the previous engine to support the concepts. The new search engine operations are divided mainly into 5 sub-systems (1) registration sub-system, (2) deregistration sub-system, (3) login sub-system, (4) search sub-system and (5) loan & reservation sub-system.

Registration Sub-System. Users will only encounter the registration interface (Figure 7) the first time they register with the system. During registration, information about the user is used to generate a profile of the user based on the W3 model shown earlier.

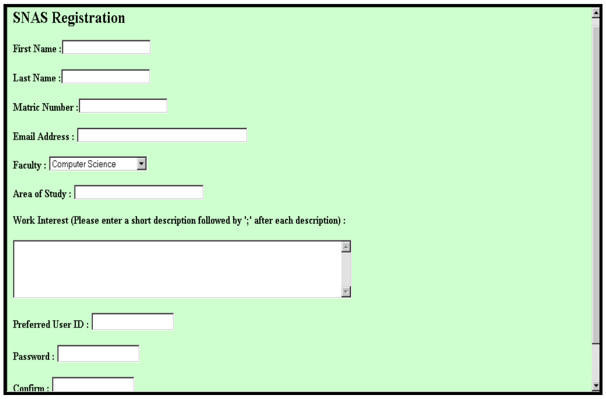


Figure 7. Registration Interface

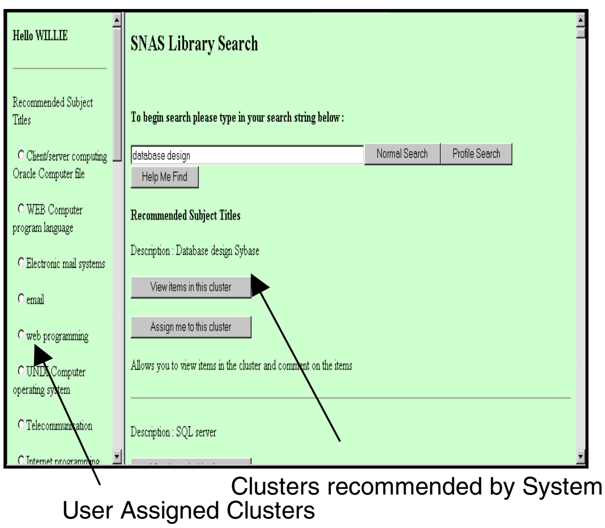


Figure 8. Screen-shot of search input

Deregistration Sub-System. This module takes care of users who have left the community and will remove all information on the user as well as update existing cluster information (i.e. if the number of users assigned to a particular cluster is 0 due to deregistration then we will remove cluster from system).

Login Sub-System. The login sub-system generates the interface for logging in and validates if the user is a first-timer or a valid user of the system. Once the user is logged in the system will begin monitoring user's interaction with the system.

Search Sub-System. The search sub-system (Figure 8) is the main interface between the user and the other modules in the system. It provides relevant records to user based on the search text (Figure 9) and on the user's profile by filtering away unwanted records. The search sub-system allows users to refine their search by providing additional keywords derived from the subject description of each item. This means that a user is able to search for similar items using the same search text coupled with the subject description belonging to a particular item. This is essential to the system as it actually masks the user feedback on the items that the user prefers. Finally, the search sub-system also provides additional information on items recommended by other users but not seen by the user based on the search text that was used to do a search.

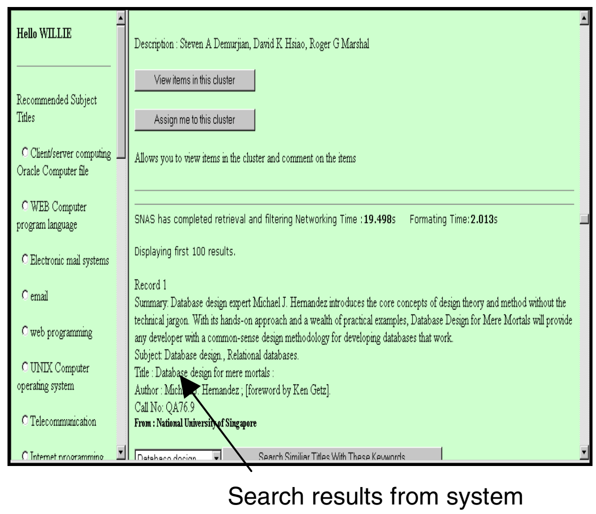


Figure 9. Screen-shot of search results

Loan & Reservation Sub-System. The search engine will only connect to the library's Z39.50 server to allow it to search for records. The loans and reservations are simulated as we do not have authorization to do so. These loans and reservations information is stored in a secondary repository or database. Figure 10 shows a screenshot of the interface.

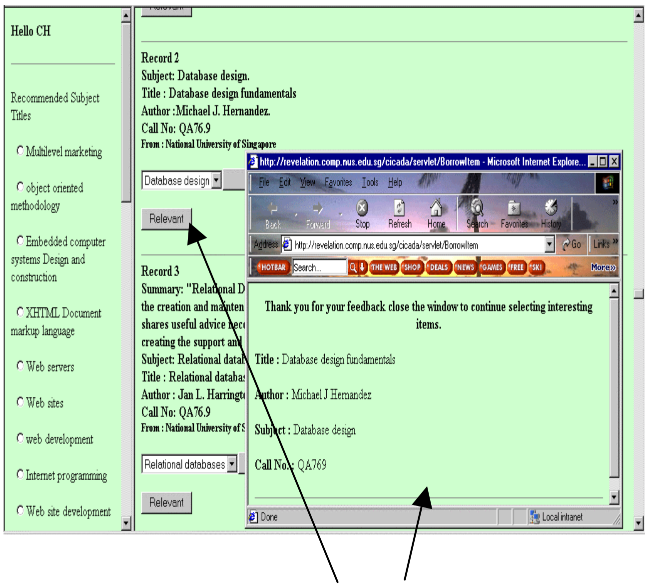
System Agent. The system agent receives user feedback from the search engine and uses the information to construct and shape the user's profile. The Dynamic Content filtering techniques incorporated into the user's profile:

1) Monitor user loans and reservation

2) Monitor user search history and

3) Monitor user keyword preferences.

Each of the techniques is set as different operations within the System Agent as shown in Figure 11. The fourth operation is to filter unwanted records from the search results base on the user's profile, the sifter operation.



Confirmation Screen appears when user clicks on relevant button simulating a loan or reservation

Figure 10. Loan and Reservation Sub-system interface

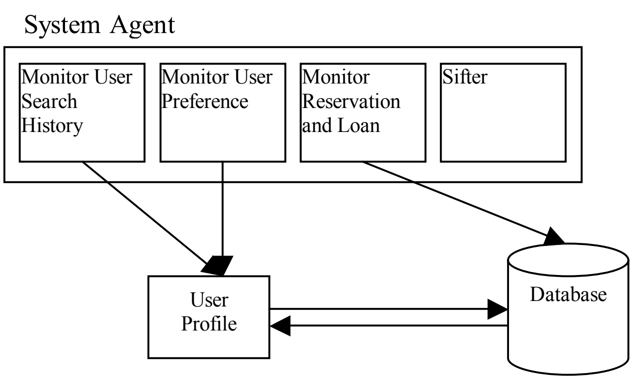


Figure 11. System Agent

Profile Manager. The Profile Manager (Figure 12) consists of three modules 1) Data Miner 2) Profile Matcher and 3) Search Assistant each performing one of the above operations respectively. The basic function of the data miner sub-module is to create and destroy clusters based on the loans and reservations that the users make. It also assigns users to relevant clusters based on the same data and maintains cluster data in the database namely, the life cycle of cycle clusters and items in each cluster. The other operations are profile matching and search notification. Profile matcher matches a new profile with existing profiles and the search assistant notifies users of new results through email even after they have logged out.

Given that each module in the Profile Manager independently accesses the database to retrieve and modify user and system data, a multi-threaded design is needed to consider the issues of data integrity.

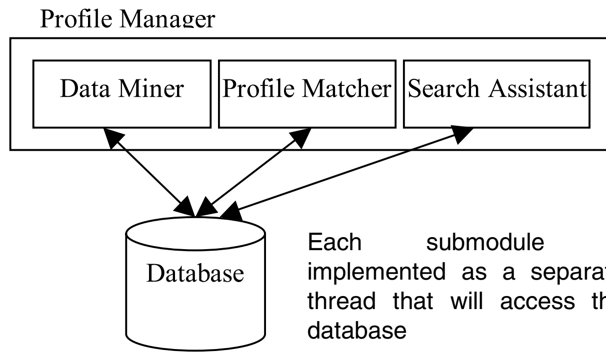


Figure 12. Profile Manager

This manager is implemented as a background daemon process. Each module is implemented as individual classes and execute as threads.

5. Performance evaluation

We conducted a preliminary study to determine if user profiling based on the four new concepts discussed, will help to improve precision and reduce the number of irrelevant records. The purpose of this study is to determine the potential of employing the novel concepts proposed in practical systems.

5.1. Experiment Setup

In the experiment, twenty participants were invited to determine the performance of SNAS. We divided the users into four different user groups so that we can employ collaborative concepts. Each group has a common objective. Each user group consists of five individuals where each individual assumed a role. Thus, every user in each group has a common task or objective. For instance, one of the user groups was asked "to build a web portal to host user forums and chat groups". Each user in the group was assigned different roles, for e.g. "Systems Engineer", and was given a description of their job scope. Each user was asked to use the search facility based on their roles assigned to them during the

0-7695-1874-5/03 $17.00 © 2002 IEEE

experiment. The experiment was conducted in two stages and in each stage the user was asked to search using the same search query for each stage. Stage 1 consists of searching using SNAS without employing any form of profiling and Stage 2 consists of searching using SNAS with the use of our proposed profiling concepts. At each stage, the users were asked to record the search query, the number of relevant records returned and the total number of records returned by the search engine. Instead of assigning different users into two groups (one using SNAS with profile and one without SNAS profile) users were asked to use the same set of keywords for both Stage 1 and Stage 2 to complete the same task. Hence, there was no learning involved in the process. In Stage 2, the users are required to fill in a registration form which consists of questions related to the role they are undertaking. The static profile is built using the user information obtained. After filling the form, they begin their search using keywords used in Stage 1. The dynamic profiles are built based on users' actions such as the records that the users marked relevant as described in Section 3.2.

Based on the responses collected from the experiments, we defined two measures to evaluate the performance of our approach. As the total number of relevant records in the system is not known, we use relative precision, P as a means to define our metric. The first metric, I_p, measures the improvement of relative precision and the second metric, I_o, determines the improvement in reducing information overload. Relative precision P, is defined as

$$P=N_r/N$$

where N_r is the number of relevant records retrieved and N is the total number of records that are retrieved. Hence, I_p and I_o are defined as follows:

$$I_p = P_{profile} - P_{no\ profile}$$
$$I_o = N_{profile} - N_{no\ profile}$$

5.2. Experimental Results

On the whole, users prefer searching using SNAS with profiling. Figure 13 shows the relative precision of each user in each stage. From the figure, we see that the relative precision values are generally higher for each user when they search using SNAS with profile than a search using SNAS without profile. There is an exception for user 16 as the relative precision was 54.1 without profile and 37.8 with profile. This was because the user entered very specific queries which were very different from other subjects. Thus, the system was unable to recommend good terms as it could not match any suitable profile of other users.

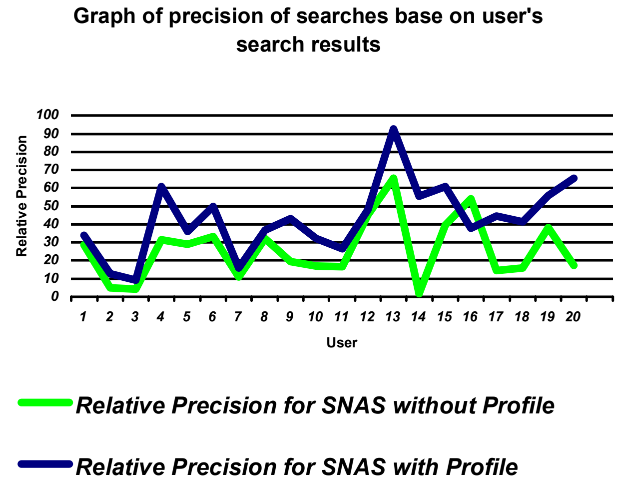


Figure 13. Graph showing precision for individual users

We believe that there is potential in applying the four concepts that we have identified as the majority of the users have benefited from the system but a more thorough experiment can be done. The averaged results are shown in Table 1. This might be due to the better search query terms that have been used to expand their queries that is generated in accordance with the user's interest profile and making use of the information from other user's with similar profiles. From the experimental results, there is an improvement of 17.1% improvement in relative precision with the use of our approach of user profiling despite having an average reduction of 111 records using SNAS.

Table 1. Average Results

Method	Average Precision (%)	Average Number of records retrieved
Search Engine with SNAS	43.0	310
Search Engine without SNAS	25.9	199
Improvement with SNAS	17.1	-111

6. Conclusions

In this paper, we have presented SNAS, an information retrieval system that employs new concepts that we have proposed, dynamic content profiling, dynamic collaborative profiling, static content profiling and static collaborative profiling. From our experiment, we believe that user profiling using these new concepts that we have highlighted in this paper have the potential to alleviate the problems encountered by users who are overwhelmed by the amount of information retrieved by various search facilities.

7. References

[1] F.A. Aniscar and C. Tasso. "ifWeb: a Prototype of User-Model-Base Intelligent Agent for Document Filtering and Navigation in the World Wide Web", *Proc. of the workshop Adaptive Systems and User Modelling on the World Wide Web, 6th International Conference on User Modelling UM97*, 1997.

[2] C. Avery and R. Zeckhauser, "Recommender systems for evaluating computer messages", *Communications of the ACM*, 1997, 40(3):88-89.

[3] C. Bradford and I. Marshall. Analysing Users WWW Search Behaviour. *Proc IEE colloquium 99/149* - navigating the web.

[4] M. Ester, H. Kriegel, J. Sander, M. Wimmer and X, Xu, "Incremental Clustering for Mining in a data warehousing environment" *Proceedings of VLDB Conference* 2000.

[5] T. Hirashima, N. Matsuda and J. Toyoda, Context Sensitive Filtering for browsing in hypertext, *Proceedings of IUI98 (Intelligent User Interfaces)*, p. 119-126, 1998.

[6] L. Kelly and J. Dunnion, "INVAID: an intelligent navigational aid for the World Wide Web", *IEEE Two-day Seminar. Searching for Information: Artificial Intelligence and Information*, p. 14-17, 1999.

[7] J.A. Konstan, B.N. Miller, D. Maltz et.al., "Grouplens: Applying Collaborative Filtering to Usenet News", *Communications of the ACM*, 40(3):77-87, March 1997.

[8] S. C. Newell, User Models and Filtering Agents for Improved Internet Information Retrieval, *User Modelling and User-Adapted Interaction* Vol 7: p. 223-237, 1997.

[9] D. Oard, Information Filtering Defined, 1995, University of Maryland Department of Electrical and Computer Engineering, http://www.enee.umd.edu/medlab/filter/filter_definition.html [Online]

[10] M. Pazzani, J. Muramatsu, D. Billsus, Syskill and Webert: Identifying interesting web sites, *13th National Conference on Artificial Intelligence*, 1996.

[11] D. Poo, T. Toh and C. Khoo, Search interface for Z39.50 compliant online catalogs over the Internet. In *HICSS-32, Proceedings of the 32nd Annual Hawaii International Conference on System Sciences, Software Technology Track, Multi Media Database and Internet Mini Track (USA)*, Hawaii, Jan 5-8, 1999; New York: IEEE, 50-57, 1999.

[12] H. Ramscar, H. Pain and J. Lee, Do we know what the user knows, and does it matter? The epistemics of user modeling, *6th International Conference on User Modeling UM97*, p. 249-243, 1997.

[13] S. Stewart and J. Davies, "User profiling Techniques: A Critical Review", *19th Annual BCS-IRSG Colloquium on IR Research*, 8-9, 1997.

[14] Y.H. Wu, Y.C. Chen and Arbee L. P. Chen, Index Structures of User Profiles for Efficient Web Page Filtering Services, *Proceedings of IEEE Conference on Distributed Computing Systems*, p. 644-651, 2000.

[15] T. Zhang, R. Ramakrishnan and M. Linvy, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proceedings of ACM SIGMOD Conference*, p. 103-114, 1996.