Journal of Network and Computer Applications 35 (2012) 1446–1453

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Enhancing meta-portals using dynamic user context personalization techniques

Christos Bouras *, Vassilis Poulopoulos

Research Academic Computer Technology Institute and Computer Engineer and Informatics Department, University of Patras, Rion GR26504, Greece

ARTICLE INFO

Article history:
Received 20 January 2011
Received in revised form
1 August 2011
Accepted 10 October 2011
Available online 30 October 2011

Keywords:
Personalization
Web personalization
On-line document grouping
User profiling
Meta-portal

ABSTRACT

The Internet is flooded with information and the last decade its size has grown so many times that information search and presentation have become tedious tasks even for experienced users. Minor changes to existing resources can alter the situation and lead to major changes to the end user experience. In this manuscript we present the dynamic web personalization and document grouping infrastructure for meta-portals and the evaluation of our mechanism on a meta-portal. A meta-portal is an informational node where articles from different sources are collected and presented in a categorized and personalized manner. The web personalization mechanism is based on dynamic creation and update of user profiles according to the users preferences when browsing. In parallel a user's profile is affected by user grouping details, which are constructed by users with similar profiles. Assuming that required information, such as article tagging, keywords to categories matching and articles to categories relation is already part of the meta-portal we present a novel mechanism that can build and maintain a user profile which is formed without disturbing the user. Furthermore, we describe the real-time user-centred document grouping mechanism that is implemented to support the web personalization system and present the experimental evaluation of the whole system.

1. Introduction

The last decade can be inevitably referenced as the decade of dramatic changes to almost every aspect of our everyday life. The advances of technology are huge and the evolution of World Wide Web (Internet) can be recognized as enormous. This weird freedom that the Internet offers, attracts more and more people. More attractive is the fact that people are free to produce on-line content in an extremely easy way making thus the production of web content a trend. The Internet is a vast place of article production and it can be referenced without any doubt as a large newsletter. The problem that arises from the fact that the Internet becomes a place where the sources (media) are more than the consumers (readers) is that the customers are usually unable to locate useful information. By useful information we define the information that an user would like to be presented, without being disturbed by any other means of content.

Searching across the Internet through the wide variety of search engines could be a possible solution to the problem of locating information, but the outrageous number of results is uninviting. The search tools that exist within article's sources and the communication channels provided can be presented as a solution or even the ultimate solution; however, the user must "invent" these places before starting to use these services. Creating customized and personalized sections within web pages is another viable solution but some recent examples seem to become misleading for the plethora of different types of users that exist on the web. User personalization and user profiling seem to be the panacea of the current chaotic web status.

User personalization is usually conflicted with the term customization. The difference is vast as the customization refers to the structure and coloring of the web page, while personalization usually refers to the content itself. What we believe is that the user should be able to adapt not only the structure of a web page, but also the content that is presented. Talking about specific content, somebody can assert that the portals are taking measures towards this problem and the content is enriched with an indication about category and lately with tagging on articles. This is sufficient up to an extent but there is still much to be done in order to extend the portals so as to present user centred information. The solution could be found on user profiling and dynamic changes to the user profile according to his habits.

We present a novel mechanism for user profile construction and maintenance in meta-portals. Many worldwide known meta-portals are Yahoo[1] and Google news.[2] We enhance the operation of our meta-portal peRSSonal by providing dynamically changing

* Corresponding author. Tel.: +30 261 096 0375.
E-mail addresses: bouras@cti.gr (C. Bouras), poulop@ceid.upatras.gr (V. Poulopoulos).

[1] http://news.yahoo.com/—news from Yahoo.
[2] http://news.google.com/—news from Google.

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.10.005

user profiling features fully adapted on the user's needs and without need of any user input.

The rest of the paper is structured as follows. The next section presents the related work while the third section our system's architecture. In the fourth section the algorithmic analysis is presented and the following section includes sets of experiments. The manuscript is finalized with future work and concluding remarks on the implemented system.

2. Related work

Many efforts were presented in the latest years in order to provide a solution to the problem of user profiling within web sites or even across the Internet. There is a slight but enormous difference between user profiling (which leads to personalization) and customization of web sites. Customization is the capability that is provided to the user to alter the layout of a web site; which is the color, the font, the position of the elements, the order of the information and others. In the context of the Internet, personalization implies the delivery of dynamic content, such as textual elements, links, advertisement, product recommendations, and more, that are tailored to needs or interests of a particular user or a segment of users (Baraglia and Silvestri, 2007). Personalization techniques (Bouras et al., 2008) are an alternative, user-centric, approach to addressing the problem of information overload. The ultimate goal of any user-adaptive system is to provide users with what they need without them asking for it explicitly (Mulvenna et al., 2000).

Coming to our first statement about the difficulty of search engine usage we investigated research work that is done to the past and it is a great proof that the situation remains almost unchanged through the years. In the majority of the currently existing search engines, when different users submit the same query, the same results are returned in the same order, regardless of who submitted the query. A recent change to Google's search engine result, seems to be misleading as the same user, submitting the same query from different machines is getting different results. Obviously, it is unlikely that all the users of a search engine are so similar in their demands that a sole approach to searching fits all needs. Indeed, in terms of searching, one-half of all retrieved documents have been reported to be irrelevant compared to what the user expected (Casaola, 1998). Additionally, a number of studies have shown that a vast majority of queries to search engines are short and underspecified (Jansen et al., 2000) and different users may have completely different intentions for the same query Lawrence (2000) and Krovetz and Croft (1992).

Some important efforts towards personalization can be found in Zaiane et al. (1998) and Mobasher (2007) where it is obvious that for more than one decade the research community is trying to apply web personalization through data mining activities and generally heuristics while (Anand and Mombasher, 2005) present some of the first more "advanced" techniques of web personalization for the web2.0 that was born back in 2005. The approaches described in Huang (2001) and Srivastava et al. (2000) are of high importance in the research literature on the issue as the first one introduces a cube model for knowledge extraction about the user's behaviour and the second deals with usage patterns from web extracted data.

Kim and Chan (2008) present a robust context for personalization based on UIH which is the user's interest hierarchy that is constructed with the usage of a tree model of the user profile. Other approaches like the ones presented in Sieg et al. (2007) and Garofalakis et al. (2008) that are applying personalized features either on portals or on search procedures by utilizing semantic information of the user are also interesting as they gather information from meta-data and not only direct information from the user. Evaluation of the user models learned from the data involves the estimation of the accuracy of the models for predicting content that may be interesting to an user as well as other aspects such as explain ability of the recommendations, diversity of the recommendation set, serendipity of the recommendations, and user satisfaction (Herlocker et al., 2004). Finally, it is important to have a reference on the ongoing discussion that is focused on the part of privacy and web personalization. It is a fact that some of the constructed mechanisms are utilizing private information which is obtained without the user's consent. Extended information about the ease of use of privacy and web personalization can be found in Wang and Kobsa (2007) where the formula for reconciling both is presented and analysed.

3. Architecture

The architecture of the system relies on distributed components which form the dynamic web user profiling system. We are putting the focus on the personalized profiling subsystem. We are also doing brief analysis of the other modules in order to cross-connect the features of our complete system, peRSSonal.[3]

The architectural schema consists of a series of subsystems, as depicted in Fig. 1. The collaboration between the distributed parts is based on the open standards (for input data and output data) and on the communication with a centralized database. The general procedure is as follows: at first, web pages are captured and only the useful text is extracted from them. Then, the extracted text is parsed in order to extract keywords and metrics while this procedure is followed by summarization and categorization. Finally we have the presentation of the personalized results to the end user.

3.1. Flow of information

In Fig. 2, we can see the general schema and flow of the advanced and personalized profiling system.

The personalization procedure of the portal that is supported as a medium of communication between all the procedures and the users can be used in order to personalize the summarization on each user. According to the algorithmic procedures of the personalized portal, the system creates a vector that represents the user's profile. To be more precise, each user has two vectors for his profile: a "positive" vector and a "negative" one. The positive vector represents semantically the interests of the user on the article content and the negative represents what is out of user's interest. The vectors are constructed from tables with keyword/value pairs. According to the user's behaviour when browsing the meta-portal the vectors are dynamically altered. The main factors that affect the user's behaviour are depicted in Table 1.

3.2. Document grouping

The system that we are presenting is utilizing the user's behaviour in order to achieve enhanced document grouping. The document grouping procedure of the system leads to creating sets of articles that are identical. By identical, we define the articles that refer to exactly the same fact but have different sources. The document grouping procedure is a never ending procedure because articles occur every 5 min (execution time of

[3] http://perssonal.cti.gr/perssonal/—peRSSonal meta-portal.

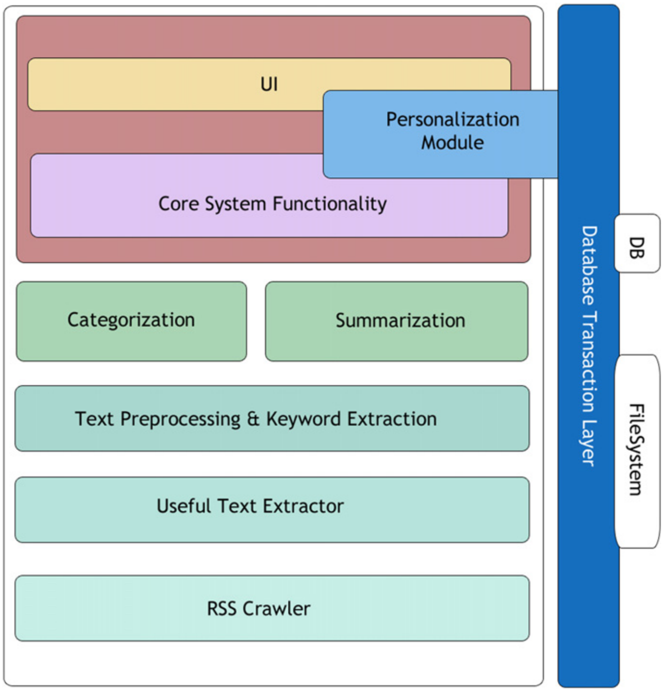

Fig. 1. System architecture.

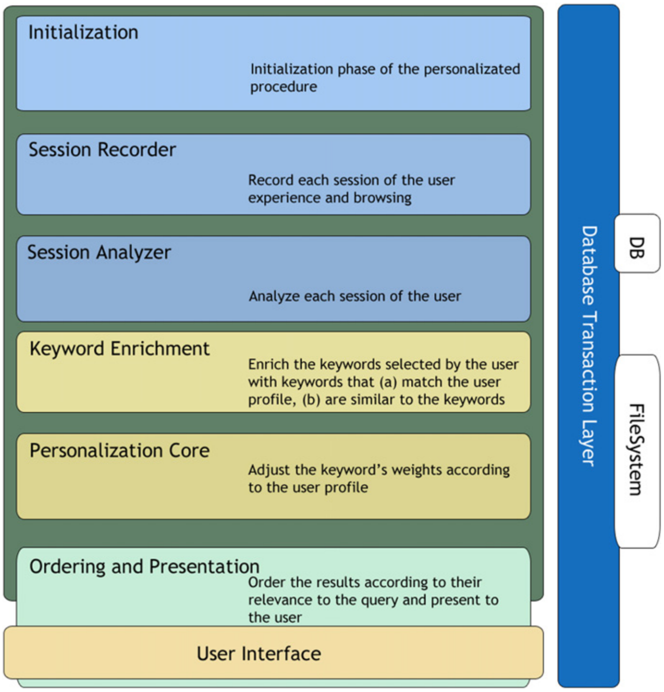
Initialization
Initialization phase of the personalizated procedure
Session Recorder
Record each session of the user experience and browsing
Session Analyzer
Analyze each session of the user
Keyword Enrichment
Enrich the keywords selected by the user with keywords that (a) match the user profile, (b) are similar to the keywords
Personalization Core
Adjust the keyword's weights according to the user profile
Ordering and Presentation
Order the results according to their relevance to the query and present to the user
User Interface
Database Transaction Layer
DB
FileSystem

Fig. 2. User profiling flow of information.

our crawler) and the groups should be constructed and maintained simultaneously. The basic idea of the document grouping procedure is the on-line creation of the document groups. When the user selects an article to read the system checks if a cluster exists for the specific article. In case the cluster exists then all the documents of the cluster are fetched and presented to the user as a single fact with different instances/sources. From the behaviour of the article publishing procedure we assume that articles published with time difference greater than 16 h cannot be identical. Still, the system can recognize such articles as relevant. This means that if the oldest article in a cluster is more than 16 h old then the cluster is considered to be "closed".

Table 1
Factors affecting user behaviour.

Factor	Influence
Selecting or not selecting an article	Medium
Position in the page of article selected or not selected	Normal
Time spending reading	
The article in combination to the article's length	Medium/High
Selection of similar articles	Medium/High
Selecting to browse the original web page of an article	Medium
Adding the article to the "starred" articles	High
Utilizing the document-clustering service on an article	Medium/High
Using the 'tag the article' service	High
Removing the article from the article's list	High
Using the tracking service	High

3.3. *PeRSSonal*

PeRSSonal, as explained in Bouras et al. (2008), follows a classic *n*-tier architectural approach. The system consists of multiple layers which work autonomously and collaborate through a centralized database. The web interface handles the information flow into the mechanism which is then directed to the interior subsystems. Text preprocessing techniques follow and the results are led to the next level of analysis where core information retrieval (IR) techniques take place. Finally the outcomes are presented to the end users through the information presentation subsystem. The collaboration between the distributed systems is based on open standards utilizing XML for input and output which are supported by each part of the system and by the communication with a centralized database.

The procedure of the mechanism as depicted in Fig. 3 is (a) capture pages from the Internet and extract the useful text (text containing the article's body), (b) parse the extracted text and preprocess it, (c) summarize and categorize the text and (d) personalize the results and present them to the end user.

In order to capture the pages, a simple focused web crawler is used. The crawling procedure is distributed across multiple systems which communicate with the centralized database. During the analysis level, our system isolates the useful text from the HTML page. More information about this procedure can be found in Bouras et al. (2005). The second analysis level receives as input XML structured information, deriving either from the database or from raw XML files, which include the article's title and body. Its main scope is to apply text preprocessing algorithms on the article, providing as output keywords, their location into the text and their frequency of appearance in it. These results are necessary in order to proceed to the third analysis level. Information about our preprocessing mechanism can be found in Bouras et al. (2006).

4. Algorithm analysis

The algorithm that fetches the article is very simple and is based on the fact that every web portal includes a series of RSS feeds that are offered to the end user. Opposed to having to visit every page of the many news portals that exist on the Internet, we fetch their RSS and more specifically the one that includes the daily "top stories". From the XML structure of the feeds we can obtain the most important articles that are published to each news portals, together with information that have to do with the title of the article, its exact URL and the date of publication. After the articles' URLs are extracted from each feed, the focused

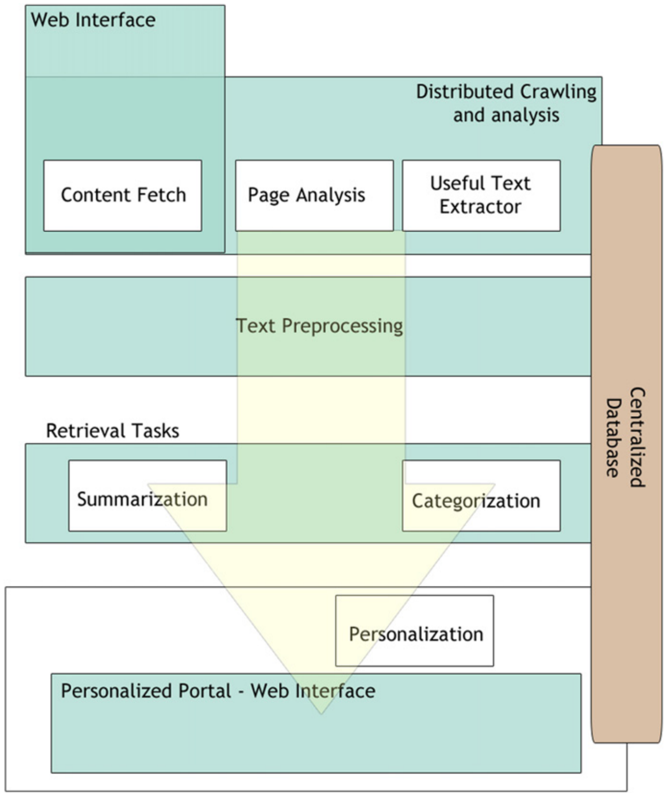

Fig. 3. PeRSSonal's architecture.

crawler, currently working as a wrapper, changes its functionality to a simple crawler and visits every singe URL extracted from the RSS feeds in order to obtain the HTML pages that include the latest articles of the RSS.

The next procedure includes the steps of analysing the HTML files, extracting the useful text from them and preprocessing the useful text in order to extract the article's keywords. The useful text extraction is based on the fact that HTML pages can be depicted as a tree with every HTML tag holding a node on the tree, while every leaf includes pure text. In order to extract the useful text we utilize a clipping extraction technique described in Bouras et al. (2005). The preprocessing techniques and the keyword extraction techniques (Bouras et al., 2008) follow. If the categorization procedure fails to categorize a text, meaning that the levels of similarities between the article and the categories provide an obscure choice, then a simple assumption is used in order to retry the categorization step. A well summarized text that includes only the important information of a text and thus, only the important keywords of it, has a higher possibility to be categorized than the original text. After a failure of the categorization procedure, the system summarizes the text and attempts a second categorization. If the categorization fails again the text is labeled as uncategorized or general.

At this stage we assume that we have all the prerequisites in order to construct and maintain an user's profile. The user's profile is created in a single step and it is maintained while the user is logging in the meta-portal and utilizing the information and services presented to him.

4.1. Dynamic user profile creation and maintenance

The user profile is created when the user is registered and it is maintained while the user utilizes the services of the meta-portal.

Some first information is obtained by the system when the user is registered in order to create an initial profile. This procedure is done through the web registration procedure. During this procedure the user is asked to enter his preferences against the seven major categories of the portal (business, entertainment, health, politics, technology, education, science). The preference varies from -5 to $+5$ indicating total reject of the category to total accept respectively. For each of the categories a vector exists in the database constructed from pairs of word roots/relevance; each pair indicating the representative word roots of the categories and the quantity of relevance to the category. Table 2 indicates the top-5 word roots/values of the category business, where the relevance derives from the tf-idf weight of the word root into the set of documents of the category. If for example the user has selected to see articles labeled as "business" when registering to the web environment then Eq. (1) is utilized in order to construct a simple vector for the user with word roots and relevance.

$$\beta(x) = \sum_{\kappa=1}^{n} \beta\kappa_x(\kappa) * \epsilon(\kappa) \quad (1)$$

where β will be the relevance for word root x, $\beta\kappa$ (m) is the relevance of word root x in category κ and $\epsilon(\kappa)$ is the user's selection against a category. As the selection varies from -5 to $+5$ Table 3 indicates the value of ϵ according to the user's choice. The algorithm for the definition of ϵ is

$$\epsilon(\kappa) = 3 * X_\kappa^2 \quad (2)$$

This is done for the first $\epsilon(\kappa)$ word roots of each of the category (κ) as they are considered to be the most representative in order to create an initial profile for the user. For a category with preference $+4$, the 64 most representative keywords are selected. Finally, the user ends up with a profile that consists of a single vector constructed from at most 700 pairs of word roots/values. This vector is the initial vector that is used to present the first list of articles considered to be similar to the user's profile and consists of terms that can be positive or negative. The positive terms are used in order to obtain the articles that are relevant to the users and the negative are used in order to reject articles from the ones that are selected for the user. The algorithm that is used in order to measure the relevance of an article (document) to the user (terms—query) is a variant of the Okapi BM25 set of

Table 2
Top-5 word roots/relevance for "business" category.

Word root	Value
Price	0.029
Compan	0.028
Market	0.028
Bank	0.027
Rate	0.027

Table 3
Value of ϵ according to user's choice (positive only).

Selection	Value of ϵ
±5	±100
±4	±64
±3	±36
±2	±16
±1	±4
0	1

Table 4
Changing the weight for updating user's profile.

Action and weight (percentage)	
Read similar articles	10
Load tagging	10
Track article	20
"Star" article	20
Load original page	10
Remove article	−60

algorithms (Jones et al., 2000) which is out of the scope of the current work.

For the maintenance of the user profile a set of algorithms is utilized which derive from the usage of services of the meta-portal. The scope of each of the algorithms is to add new pairs into the user's vector or update the existing ones, and the information on how to do this derive from the user's activities in the meta-portal. Each article consists of pairs of word roots/relevance. Experimental evaluation that was done into news articles has shown that the first 10 keywords can represent fully the article's semantic quality. The information that is collected in order to update the profile of the user is the articles that an user has read, the articles that are rejected while all the other information are collected while the user is reading an article. When the user starts a session, a session recorder is responsible for recording all the user's input. A weight is assigned to every keyword/relevance existing in any article that an user reads or rejects. When the user is reading an article an initial state of the weight is given according to Eq. (3).

$$weight(keywords) = \frac{min(timereading(x),time2read(length(x)))}{time2read(length(x))} \cdot \left(1+\frac{articleposition(x)^2}{\sqrt{articleposition(1)^2+articleposition(2)^{+}articleposition(n)^2}}\right) \quad (3)$$

After opening an article to read it the weight can still be changed. Table 4 shows how this weight is changed. Each action is recorded and when the article page is unloaded the sum of the weight percentage to be added to the pairs is added directly to the weight deriving from Eq. (3). At the end of the session, each keyword that was located in one of the documents read or in one of the documents rejected is followed by a weight either positive or negative. The system checks if this word root already exists in the user's vector. If the word root exists then the system updates the value of the word root into the user's profile by directly adding the weight to the already existing one without exceeding the triple value. If the word root does not exist into the user's vector then a new entry is added with a value equal to the weight that was recorded but not larger than the double of the maximum existing value. The limits exist in order not to overload the user's profile quickly with word roots but form a user's profile gradually. In case of negative weights we apply the same procedure with the same limits.

4.2. On-line document clustering

The on-line document clustering is a procedure that takes place while an user is reading an article and its scope is to collect and interconnect every document that is identical to any other document but they derive from different sources. This is done in order to omit any duplicate instances of the articles, create linkages between articles and in parallel present at once to the user every document that concerns an event or a new. Moreover, this procedure speeds up the presentation of sets of articles to the end user. The algorithm that is utilized in order to locate all the identical articles is based on the cosine similarity measure and is done real-time. In order to present how the on-line document clustering is done we are making two basic assumptions: (a) the system has never done document clustering before and (b) a document can be related to any other document if they have three days difference at most. When the user is selecting an article to read, an function, that relies on AJAX technology, is responsible for fetching the cluster of documents that are directly related to the document that the user is reading. The following pseudocode presents the steps for locating the identical articles.

```
program Identical
  const
    current_article;
  var
    article(three days difference);
  similarity: Real;
  begin
    foreach(article)
  similarity=cosine_similarity(current_article,
article)
    if(similarity > =90%)
    //big enough to indicate identical articles
      if(has_cluster(article))
        add2cluster(current_article)
        break; //terminate all procedures
      else
        create_cluster(current_article, article)
      end if
    end if
  end foreach
end.
```

The whole procedure is done asynchronously without distracting the user. If the document cluster already exists then all the articles within the cluster are directly presented to the user. In parallel, even if the cluster exists, and because of the fact that articles are added every five times, if the newest article in the cluster is not older than three days the system keeps checking for articles that may belong to the current cluster. If an article that the user is reading does not belong to a cluster then the cluster is created while the user is reading. The user is involved in the procedure of cluster creation in order to assure that the cluster consists of identical articles only. A slight change to the aforementioned pseudocode makes the difference. Assuming that an user is presented an article this indicates that the user is interested into reading the article, which furthermore means that the vector of the user's profile is close to the vector of the article. What we expect is the identical articles, that are presented to the user, to be close to the user's vector (of the user that are presented to). The limit that we have with the addition of the user's interaction is another limit that has to be passed except from the similarity of the documents. We furthermore expect the articles that will form a cluster to have similar relation to the user. If the article that the user is reading has similarity A with the current user then the rest of the articles that are meant to form the cluster should have $\pm\beta*A$ where β varies from 0.07 to 0.1 according to our experimental evaluation and it is directly dependant on A. If A is relatively small (less than 30%) then it seems that the limit of β should be 0.1 while when A reaches values of 80% or more then β could be 0.07. It seems that the use of the median (0.085) is sufficient taking into account that most of the articles that are presented to an user have usually 50% relation to the user profile's vector.

5. Experimental evaluation

The experimental evaluation of the system consists of experiments conducted in order to present the creation and maintenance of the user profile and to provide information about the statistics of the document grouping procedure. We utilize peRSSonal meta-portal which we enhance with the dynamic user profiling mechanism and the document clustering subsystem, and we are executing our experiments on both real and virtual users that are registered officially in peRSSonal. In parallel we are presenting an analysis on the overhead of the profile construction and maintenance procedure.

5.1. Experiments on the dynamic user profile

In order to conduct experiments on the dynamic user profile we are first experimenting with the reason of existence of a user profile within a meta-portal. Figures 4 and 5 present how many articles interest a user without profile from the ones that are presented to her/him and what is the corresponding fraction when user profiling and personalization is used.

It is obvious that when an user enters and browses the meta-portal without creating a profile the articles that are presented to her/him concerning one category of the portal are usually more than 100. From them, the user normally selects 7–15 articles to read. It seems that only a 10% of the articles presented are of the user's interest. On the other side, when an user creates a profile the articles presented to her/him concerning one of the categories that the user has selected are not more than 15. From the articles selected more than 75% are selected by the user to be read. The difference is huge and implies that the personalization is essential for a meta-portal that presents huge amounts of information. The results extracted from the current experiment is that the system is able to create a user profile and adapt on the profile of each user. It is clear that a system supporting user profiles when presenting information can have extremely different results even on the psychology of the user. This is affected by the fact that the user is not bombarded with a vast amount of uninspiring information but only with content that concern him or her.

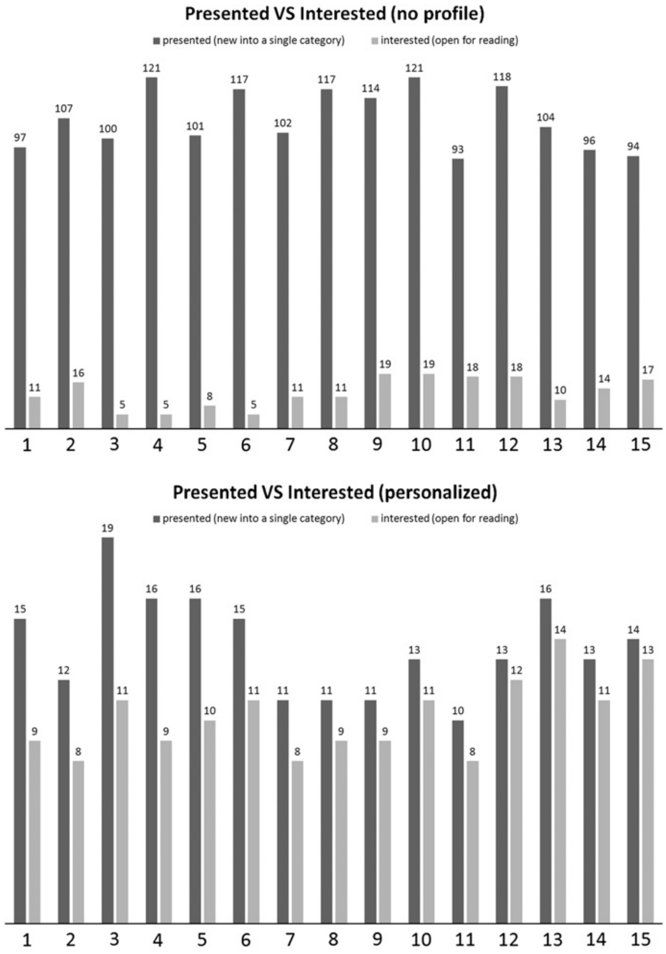

Fig. 4. Presented vs. interested without profile. Presented vs. interested with use of dynamic profile.

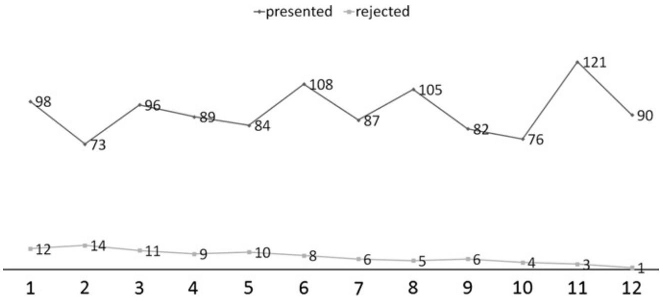

Fig. 5. Presented vs. rejected (per week).

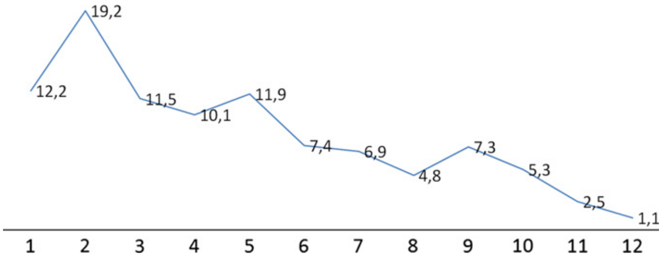

Fig. 6. Percentage of articles rejected (per week).

Another set of experiments is conducted with the help of an add-on to the meta-portal in order to obtain information about how much time is required for an user in order to create a sufficient profile and present only information that are of high interest to the user. The adaptability of the dynamic profile mechanism can be measured by asking the testers of the system to record how many of the weekly presented articles they reject.

As it is obvious after four weeks of the user browsing the meta-portal more than 90% of the articles presented to the user are of the user's interest. This is another confirmation of the system's ability to present accurate information to the end user.

Moreover, it is important to examine the percentage of articles rejected per week in order to calculate the rate of adaptation of the system to the end-user. In Fig. 6 we can see that after 12 weeks only one article, from the 100 presented, is rejected from the user. This proves that the system can locate the behaviour of the user and decide which articles to present and which not.

5.2. Experiments on the document clustering

In order to test the document grouping mechanism we conducted a basic experiment to test its efficiency. We searched through the major portals that the system checks to find manually the same article published in all of them. After ensuring that our meta-portal

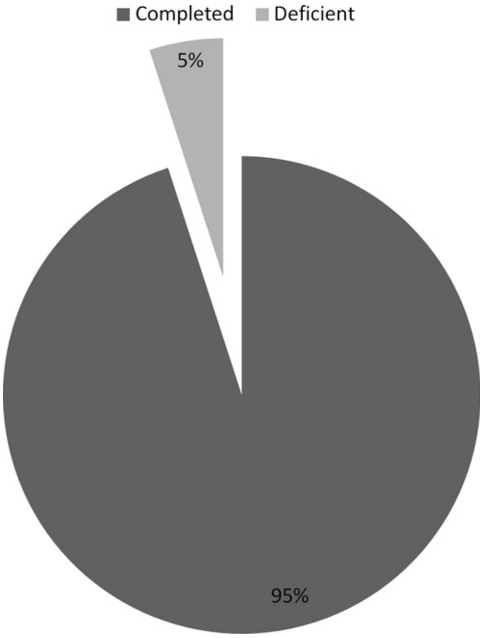

Fig. 7. Completed vs. deficient clusters—analysis of deficient document clusters.

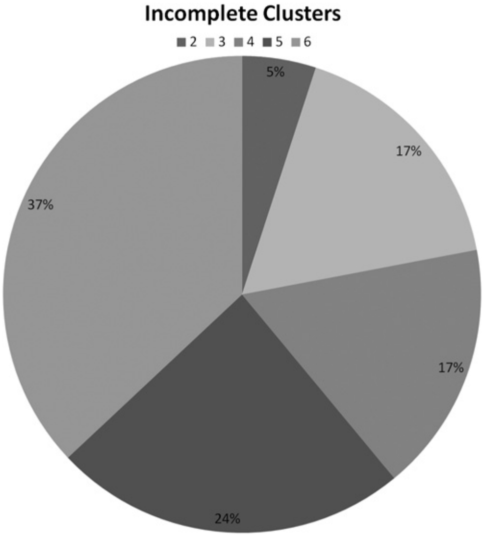

Fig. 8. Analysis of deficient document clusters.

has obtained all these articles we open one of them to see if the system is able to construct a linkage between all of them. For this reason we are checking the RSS feeds of politics news of seven major portals of Europe and the USA. When we locate an article that is published to all of them we check our meta-portal and open one instance of the articles and check if all the other six instances (one shown and six more articles from the seven portals) are present. Figure 7 presents the efficiency of the mechanism against 7336 articles (7 × 1048 articles). We expect the system to construct 1048 distinct document clusters.

We furthermore analyse the 54 clusters that fail to include the seven instances of the articles in order to see how many instances they were able to include. Figure 8 presents the results. It is clear that the vast majority of the incomplete clusters include at least five of the seven published articles (more than 60%).

It is clear that the vast majority of the incomplete clusters include at least five of the seven published articles (more than 60%). We prove that the system can achieve a dynamic document connection that is really helpful especially when combined with the presentation layer to the users. The users have the ability to browse through similar articles simply and search for information is done automatically for them.

5.3. Performance analysis

A large part of the systems that apply user personalization, and more specifically those relying on Internet user profiling, face often performance issues. This is due to the fact that originally personalization used to derive from server log analysis, an offline procedure that consumes computer resources (mainly CPU and memory).

The approach that we are presenting is completely different. The profiling subsystem is implemented utilizing server side scripting and client side scripting languages (the current implementation is using PHP, Javascript and XML). The profile of the user is recorded in real time to an application readable format (selection between XML, JSON data or directly to a DBMS). This is done with direct asynchronous server requests while the user is browsing a web page (AJAX technology used). The overhead to the server is an extra request per page viewed. This procedure is non-blocking, which means that the user is unable to stop it. Additionally, the important part of session analysis and user profile update can be done in two stages. Either the profile is updated with asynchronous parallel server requests when the user is browsing a page or this can be done with offline procedures. As it is obvious from Fig. 9, while the session recorder is running after the page finished loading? the session analyser and profile updater are attempting requests for update. As this procedure can be done offline, the updater attempts small size requests for partial profile update while the user is idle and if this

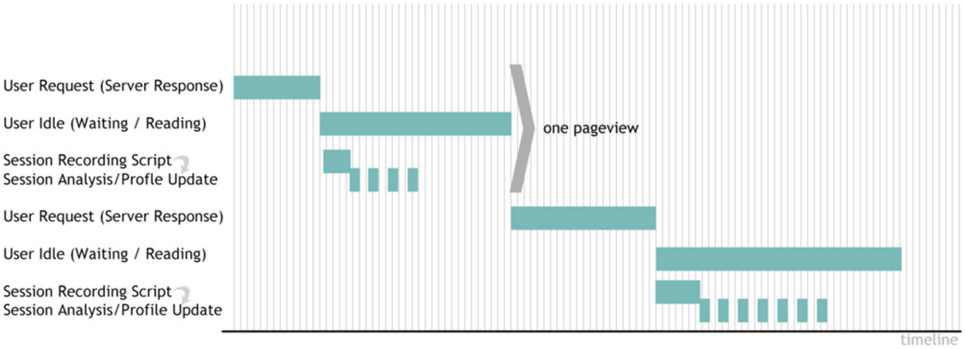
User Request (Server Response)
User Idle (Waiting / Reading)
Session Recording Script
Session Analysis/Profle Update
one pageview
User Request (Server Response)
User Idle (Waiting / Reading)
Session Recording Script
Session Analysis/Profile Update
timeline

Fig. 9. Server requests for session recording and profile update procedures.

procedure does not finish the profile update (parse of all data recorded during a session) then the procedure is finalized offline. In most of the cases, and as the user remains idle more than 10 s (which is the maximum execution time of an extremely complex user action recorded), the update of the profile, is done in real time. In this way, the consumption in server resources is as high as a webpage request is.

6. Conclusion and future work

In this paper we presented a mechanism that is able to complete a procedure of collecting news from news portals and blogs and present them personalized back to the end-users by applying furthermore document clustering algorithms. This mechanism is helpful for Internet users who are spending a considerable amount of time trying to locate news of their interest through major or minor news portals or even through RSS feeds (RSS readers). Despite the fact that the personalization micro-sites that exist, even within some portals, resolve part of the problem, still the refinement of the results and the personalization on the specific device of the user and the specific needs of the user is a huge problem. The procedure of accessing all the news portals in order to collect useful information is part of our everyday life, though, the information that is shown to the screen of the end user includes almost 80% of not needed information or even trash information. The mechanism that we are proposing is able to collect the articles from news portals (through their RSS feeds), categorize the articles, summarize them and finally present them to the end-users according to their preferences.

As an extension for our mechanism we are thinking of a news tracker system which will be able to track the changes that are done on news articles and update accordingly the document clusters. As more and more articles about a specific theme are published on several news portals or even on the same news portal we should be able to collect all the similar news and present them as one to the end user, providing also with the several links that the articles derive from and let the user make the best choice on which link to follow. Additionally, the automated procedure of maintenance of a user profile can be enhanced with user grouping procedure that will let users with similar interests exchange information on news articles. Finally, as the system is able to work at a very high speed, creating dynamically RSS for the user in real time, we are thinking of creating an add-on for every news portal that will enable the real-time creation of personalized RSS feeds for the end-user directly through the news portals.

References

Anand SS, Mombasher B. Intelligent techniques for Web personalization. in: Lecture notes in artificial intelligence, vol. 3169. , Berlin, Germany: Springer-Verlag; 2005. p. 1–36.

Baraglia R, Silvestri F. Dynamic personalization of web sites without user intervention. Communications of the ACM 2007;50(2):63–7.

Bouras C, Kounenis G, Misedakis I, Poulopoulos V. A web clipping service's information extraction mechanism. In: 3rd international conference on universal access in human–computer interaction. Las Vegas, Nevada, USA; 22–27 July 2005.

Bouras C, Dimitriou C, Poulopoulos V, Tsogkas V. The importance of the difference in text types to keyword extraction: evaluating a mechanism. In: Proceedings of the 2006 international conference on internet computing & conference on computer games development, ICOMP 2006. Las Vegas, Nevada, USA; 26–29 June 2006. p. 43–9.

Bouras C, Poulopoulos V, Tsogkas V. PeRSSonal's core functionality evaluation: enhancing text labeling through personalized summaries. Data and Knowledge Engineering Journal 2008;64:330–45.

Casaola E. Profusion Personal Assistant: an agent for personalized information filtering on the Internet. Master's Thesis. The University of Kansas; 1998.

Garofalakis J, Giannakoudi Th, Vopi A. Personalized web search by constructing semantic clusters of user profiles. In: Proceedings of the 12th international conference on knowledge-based intelligent information and engineering systems. Zagreb, Croatia; 2008. p. 238–47.

Herlocker JL, Konstan JA, Terveen LG, Riedl JT. Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems 2004;22(1): 553.

Huang Z. A cube model for web access sessions and cluster analysis. in: Proceedings of the 7th ACM SIGKDD international conference on knowledge discovery and data mining; 2001.

Jansen BJ, Spink A, Saracevic T. Real life, real users and real needs: a study and analysis of user queries on the Web. Information Processing and Management 2000:207–27.

Jones KS, Walker S, Robertson SE. A probabilistic model of information retrieval: development and comparative experiments (parts 1 and 2). Information Processing and Management 2000;36(6):779–840.

Kim H-R, Chan P. Learning implicit user interest hierarchy for context in personalization. Applied Intelligence 2008;28(2):153–66.

Krovetz R, Croft BW. Lexical ambiguity and information retrieval. Information Systems 1992;10(2):115–41.

Lawrence S. Context in Web search. IEEE Data Engineering Bulletin 2000;23: 25–32.

Mobasher B. Data mining for web personalization. in: The adaptive web: methods and strategies of web personalization. Lecture notes in computer science, vol. 4321. , Berlin/Heidelberg/New York: Springer-Verlag; 2007.

Mulvenna M, Anand SS, Buchner AG. Personalization on the net using web mining. Communication of ACM 2000;43:122–5.

Sieg A, Mobasher B, Burke R. Learning ontology-based user profiles: a semantic approach to personalized web search. IEEE Intelligent Informatics Bulletin 2007;8(1).

Srivastava J, Cooley R, Deshpande M, Tan PN. Web usage mining: discovery and application of usage patterns from web data. ACM SIGKDD Explorations Newsletter 2000;1(January):12–23.

Wang Y, Kobsa A. Respecting users' individual privacy constraints in web personalization. in: User modeling 2007. Lecture notes in computer science, vol. 4511. , Springer-Verlag; 2007. p. 157–66.

Zaiane OR, Xin M, Han J. Discovering web access patterns and trends by applying olap and data mining technology on web logs. In: Proceedings of advances in digital libraries conference (ADL98). Santa Barbara, CA; 1998.