



(22) Date de dépôt/Filing Date: 1999/03/15

(41) Mise à la disp. pub./Open to Public Insp.: 1999/09/25

(45) Date de délivrance/Issue Date: 2009/09/29

(30) Priorité/Priority: 1998/03/25 (GB9806392.8)

(51) Cl.Int./Int.Cl. *G06F 17/30* (2006.01)

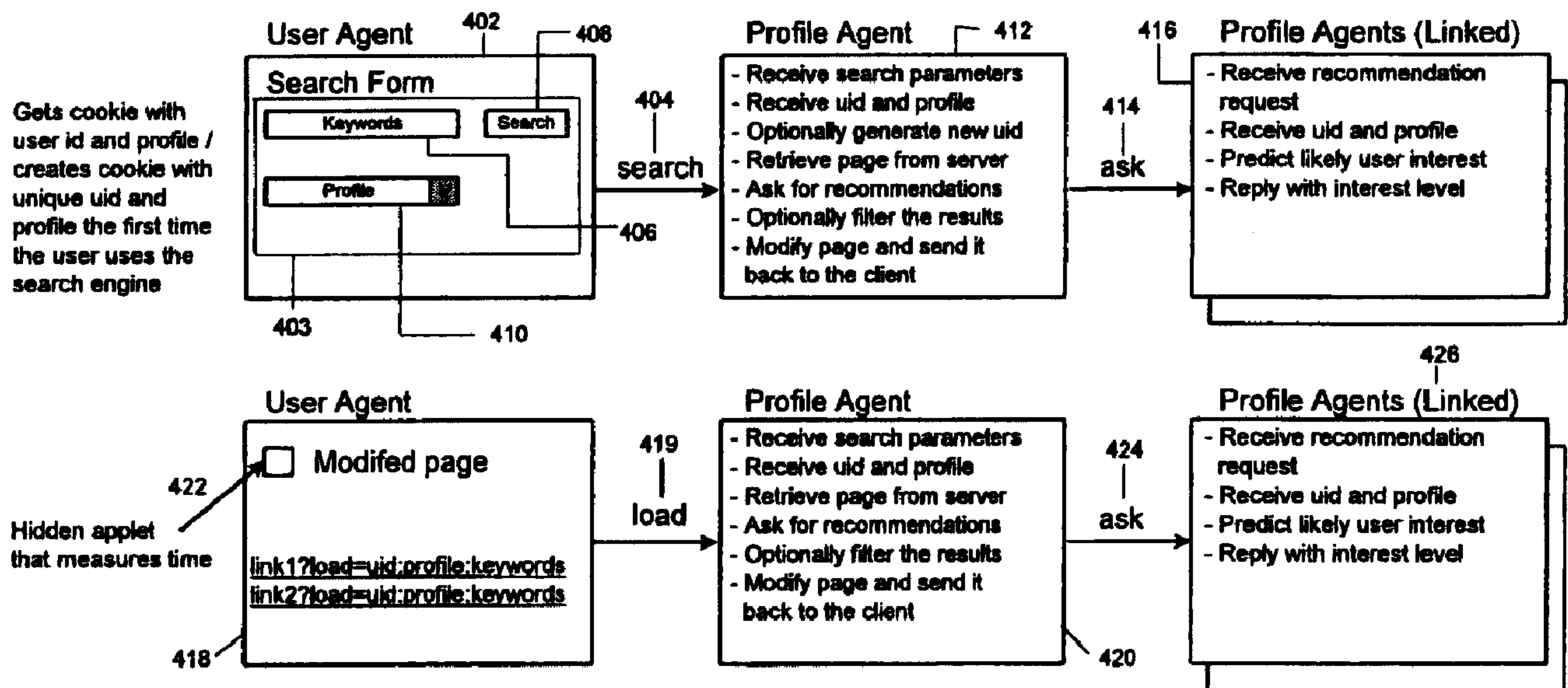
(72) Inventeur/Inventor:
WEISS, MICHAEL, CA

(73) Propriétaire/Owner:
MITEL NETWORKS CORPORATION, CA

(74) Agent: PERRY + CURRIER

(54) Titre : MOTEUR DE RECHERCHE INTERNET FONCTIONNANT COMME UN AGENT

(54) Title: AGENT-BASED WEB SEARCH ENGINE



(57) Abrégé/Abstract:

According to one aspect of the present invention, there is provided a method for searching that uses social filtering between collaborating agents to track user behavior and to guide the search for documents and information stored in electronic form. The present invention records background information about each user in a user profile. User profiles are learned from a training set of pages or by observing user behavior. The method of the present invention increases the number of search results which are likely to be perceived by the user as highly relevant because of a "peer effect". The present invention allows the use of best-first search of the Web which performs better than the depth-first or breadth-first search used in past approaches.

ABSTRACT

According to one aspect of the present invention, there is provided a method for
5 searching that uses social filtering between collaborating agents to track user behavior
and to guide the search for documents and information stored in electronic form.

The present invention records background information about each user in a user
profile. User profiles are learned from a training set of pages or by observing user
behavior.

10 The method of the present invention increases the number of search results
which are likely to be perceived by the user as highly relevant because of a "peer
effect". The present invention allows the use of best-first search of the Web which
performs better than the depth-first or breadth-first search used in past approaches.

AGENT-BASED WEB SEARCH ENGINE

FIELD OF THE INVENTION

5 The present invention relates generally to the field of information search and retrieval, and more particularly, to an agent-based Web search engine that improves keyword-based searches by utilizing a user profile to further restrict the search to documents of interest to the user. The present invention utilizes a network of collaborating agents that track and use information from the browsing behavior of other
10 similar users.

BACKGROUND OF THE INVENTION

 Previous approaches to finding information on the World Wide Web (WWW)
15 have included automated searching programs (search engines) such as WAIS or Web crawler to locate web sites and information of interest to the user.

 These automated search engines suffer from the problem of returning too many search results, frequently by including documents of marginal or
20 low relevance, reducing the usefulness of the search to the user. These typical approaches fail to consider any measure of the user interest in conducting the search. These prior art approaches use a measure of interest based only on the search keywords entered by the user. As a consequence, these search approaches return all documents which contain the search terms, including documents which are in subject areas
25 unrelated to the user's area of interest. However, frequently background information is available which can be applied to the search. These prior art approaches do not use this valuable background information to eliminate documents which are not relevant to the user. User satisfaction with these prior art search engines is therefore low.

 Lieberman in "Letizia: An Agent that Assists in Web Browsing", International
30 Joint Conference on Artificial Intelligence, 1995, describes an approach that uses a single agent to assist the user browsing the World Wide Web. In the Lieberman approach, the agent tracks user behavior and attempts to anticipate documents of

interest by autonomously exploring links from the user's current position. This approach infers search goals from the user's browsing behavior and makes unsolicited recommendations of "interesting" documents. One of the drawbacks of this prior art approach is that it focuses on the behaviour of the individual user without considering
5 other information which can be gleaned from community interest in the documents and information.

It is a reasonable assumption that documents and information of interest to one person in a community would likely be of interest to another. Valuable information obtained from the browsing behaviour of other users can be used to focus the search.
10 Therefore, an approach which tracks and uses information from the browsing behavior of other users "similar" to the user can be used to facilitate the search process to ascertain relevant documents.

Furthermore, a search mechanism which is augmented by utilizing a network of collaborating agents to track browsing behavior and guide searches would improve the
15 effectiveness of the search.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, there is provided a method for
20 searching that uses social filtering between collaborating agents to track user behavior and to guide the search for documents and information stored in electronic form.

The present invention records background information about each user in a user profile. User profiles are learned from a training set of pages or by observing user behavior.

25 The method of the present invention increases the number of search results which are likely to be perceived by the user as highly relevant because of a "peer effect". The present invention allows the use of best-first search of the Web which performs better than the depth-first or breadth-first search used in past approaches. The approach according to the present invention is also highly scalable, since each agent
30 manages only a small subset of users and documents, as compared to other known social filtering approaches (e.g. Maes, P. "Agents that Reduce Work and Information Overload", Communications of the ACM, July, 1994).

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram depicting an overview of a networked system implementing the search mechanism of the present invention.

5 Figure 2 is a block diagram of a search tree for searching web pages illustrating the use of the present invention.

Figure 3 is a block diagram depicting the present invention implemented in Java.

Figure 4 is a flowchart diagram illustrating the executing of the search commands of the present invention.

10

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

According to the present invention, there is provided an agent based web search engine where agents are used to assist users and communicate information to facilitate
15 efficient searches. The data that is communicated among agents pertains to user profiles about individual users on the web as well as web page profiles regarding the particulars of various web pages.

The present invention tracks and uses information gathered on the browsing behaviour of users that are “similar” to the user searching for information, utilizing a

process known as social filtering. The concept of social filtering has been described by Maes (referred to herein above), and by Lashkari, Y. et al. "Collaborative Interface Agents", Conference of the American Association of Artificial Intelligence, 1994.

Social filtering, in its basic unmodified form, does not base its correlations on the content of information but rather on correlations solely among the users or viewers of such information. Social filtering uses information about a user's social environment as a guide to locating relevant documents and information.

In the implementation of social filtering, information about the interests of individual users is gathered. Then, during the search phase information is filtered for relevance by exchanging data about the users who have expressed an interest in the information. Social filtering has, for example, been successfully applied to make recommendations about music records.

Turning to Figure 1, a networked system implementing the present invention is shown. Web server 110 is connected to local area network 104. Local area network 104 is in turn connected to the World Wide Web (WWW) 102. Likewise, web server 112 is connected to local area network 106, which is connected to World Wide Web 102. Web servers 110 and 112 are standard Internet or Intranet computing machines, as are well known in the art, that are capable of displaying web pages of hypertext markup language (HTML) format. HTML is a well known markup system used to create hypertext documents that are portable from platform to platform. To accomplish the communication tasks required of the present invention, web servers 110 and 112 use standard network communication schemes and the Internet Hypertext Transfer Protocol (HTTP) which allows the transfer of information from client to server. While the invention hereinafter will be described with respect to HTML web pages using HTTP, it is within the scope of this invention that other document formats and protocols could be used.

HTML web pages are stored in computer memory 114 of web server 110 and are made accessible to local users 118 and 120 as well as remote users 122 and 124 through the World Wide Web 102. Likewise, other web pages are stored in computer memory 116 of web server 112 which are available to local users 118 and 120 as well as remote users 122 and 124 through World Wide Web 102. Local users 118 and 120 and remote users 122 and 124 use a standard web browser, such as NetScape™ from

NetScape Communication Corporation or Microsoft Internet Explorer™ from Microsoft Corporation, which can read the HTML coded web pages. Each of users 118, 120, 122 and 124, as well as each web page in memory 114 and 116 have an accessible address or universal resource locator (URL) which can be used by users, agents or other network devices for locating and accessing user or web page information.

In the preferred embodiment, agents are used to perform tasks on the user's behalf, train or teach the user, help different users collaborate, and monitor events and procedures. In particular, various user and web page profiles are managed by agents. In addition, searches are facilitated by the communication and collaboration of agents. In the preferred embodiment, agents employ social filtering relying upon correlations drawn between different users in identifying relevant documents and information.

As shown in Figure 1, agent 126 manages a portfolio of user profiles for local users 118 and 120 on web server 110. Agent 126 also manages a portfolio of web page profiles for each web page stored in memory 114 of web server 110. Agent 128 manages a portfolio of web page profiles for each web page stored in memory 116 of web server 112. In addition, agent 130 manages the user profile for user 122, while agent 132 manages the user profile for user 124. Particulars of the profiles and the information managed and communicated by agents 126, 128, 130 and 132 is described in further detail below.

A user profile, such as managed by agents 126, 128, 130 and 132 is a description of background information about a particular user. It is created by identifying and listing specific features or areas of interest to the user. One method of representing this profile is as a vector where each element represents the user's interest in a particular feature. Therefore, in the general case, the Boolean feature vector $\{u_1, \dots, u_n\}$, each Boolean value u_k indicates whether a particular feature k is of interest to a user. For example, features for comparison could be defined as: (a) cars; (b) sports; (c) cooking. If a user is interested in cars and sports, but not interested in cooking, their profile would be represented as the feature vector $\{1, 1, 0\}$, where 1 represents true and 0 represents false. This profile could then be used to correlate with profiles of users with similar interests.

A web page profile is merely a list of the profiles of users who have visited that web page document. It may be implemented as a list of the actual user profiles, or

6

merely a pointer to the agent managing each user's profile. For example, in one embodiment, a web profile can be set up by an agent as a table of {feature, user id, interest} tuples.

An example of such is set out in Table 1 below, such as:

5

User id

Feature	a ₁	a ₂	a ₃
cars	1	0	0
sports	1	1	1
cooking	0	1	1
Java	0	0	0
Internet	1	0	1

TABLE 1

The web page profile is updated as a new user visits that web page, in a similar manner to which a web page hit counter is updated.

Before a search can be conducted, it is necessary to create user profiles. While user profiles could be manually generated, the preferred approach is to generate user profiles during a learning operation. During the learning operation, users will be presented with a set of web pages or questions and asked to indicate their interest by responding with either yes or no. Training pages are prepared such that they are associated with a set of features that will be included in the user profile. Therefore, given the set of positive examples of concepts (or web pages) that the user is interested in, and the set of negative examples (or web pages) the user is not interested in, the set of features that distinguish pages rated as interesting from those that are not can be learned through an information theoretic approach. The use of an information theoretic approach is known in the art.

Training pages that are highly matched with a user's interest may subsequently be suggested to the user as starting points for exploration of the web. This provides the additional advantage of directing

the user to a web server that supports the agent based web server engine.

The present invention is better illustrated by way of example. Turning to Figure 2, the search tree depicting the typical search actions of users of Figure 1 is shown in greater detail. The search is represented by numerous nodes 202, 204, 206, 208, 210, 212, 214, 216 and 218. Each of the nodes in the search tree corresponds to a Web page which may be on a different web server. The initial node 202 is a web page currently received by a user such as described in Figure 1. Branches of the tree represent hypertext links between the web pages. Each web page is associated with a web page profile that is used to track which users have visited that particular web page. When a user visits a web page 204 this is a good indication that it is of interest to that user.

A further, better measure of interest could be employed wherein the actual amount of time spent by a user reading a web page is recorded. This may be implemented by a profiler downloaded when a user first accesses the instrumented pages, as an invisible applet written in a suitable language, such as Java, that collects information on a user's web page traversals and captures usage access patterns including time spent on a particular page. In this manner, the time spent viewing a web page can be reported back to the agent at the web server by the applet. If the viewing time is in a certain range, the page is considered interesting to the user. The range is bounded by a lower threshold, set and adjustable by the user agent, below which the page is considered not interesting and optionally a corresponding upper threshold, beyond which the user is considered to have abandoned the web page, rather than a page of high interest. The particular search illustrated is started at the users' initial node 202. The user formulates a search by providing a keyword 222. The user profile 224 describing the user's background is implicitly added to the search specification. The user also has the option to leave the keyword unspecified. If this is done, the Web is searched for every document that matches the user's profile 224. Each node 202 - 208 has a page profile, however, only page profile 220 is illustrated in Figure 2.

As the search evolves, each of nodes 204 - 208 is tested on whether it includes the specified keyword. If it does, the correlation between the user profile 224 and the

page profile for the node is computed and compared against a threshold. A high-level formal description of the process by which a node is tested for match is described below.

5 With the following definitions:

$\underline{u}$ user profile, $\underline{u} = \{u_1, \dots, u_N\}$

u_i indicates whether feature i is of interest to the user

$\underline{m}$ matching vector, $\underline{m} = \{m_1, \dots, m_M\}$

$\underline{A}$ page profile, $\underline{A} = \{a_1, \dots, a_M\}$

10 θ threshold

N number of features

M number of agents or user profiles in the page profile

The correlation between $\underline{u}$ and $\underline{A}$ computes to:

$$\underline{u} \underline{A} = \underline{m}$$

15 A match occurs if:

$$|\underline{m}| \geq \theta$$

Each column of $\underline{A}$ in this description corresponds to the profile of a user who has visited that page. Rows are added to $\underline{A}$ as a user's visit is tracked by the agent in charge of that page. Alternatively, a link to the user's web page is provided. Any

20 obvious optimizations apply as to how these profiles are actually represented which are obvious to one skilled in the art. There are several standard ways to compute this correlation. One standard way to measure the "similarity" between two features vectors to compute their normalized vector product, that is, use a cosine similarity measure.

For Boolean feature vector, this measure obtains values between 0 and 1, where values 25 close to 0 indicate low, and values close to 1 high degrees of similarity, respectively.

The similarity function is:

$$\text{sim}(u, v) = \text{cosine}(u, v)$$

where u and v are feature vectors, and $\text{cosine}(u, v)$ is their normalized vector product:

$$\text{cosine}(u, v) = (u \cdot v) / (|u| * |v|)$$

30 Using this similarity function to calculate the similarity between vectors $\{0, 1, 1, 0, 1\}$ and $\{1, 1, 0, 0, 1\}$, the function is:

$$\text{sim}(\{0, 1, 1, 0, 1\}, \{1, 1, 0, 0, 1\}) = 2/(\sqrt{3} \sqrt{3}) = .67$$

Therefore, using the definitions above where $\underline{u}$ is the user profile of the new visitor, and $\underline{a}_k$ is the k -th column in the web page profile $\underline{A}$, the average similarity of the user profile $\underline{u}$ and the page profile gives a measure of the correlation between the user profile $\underline{u}$ and the page profile $\underline{A}$.

The correlation can be expressed by the formula:

$$\text{correlation}(\underline{u}, \underline{A}) = \frac{\sum_{k=1}^M \text{sim}(\underline{u}, \underline{a}_k)}{M}$$

This is further illustrated using the web page profile in Table 2 below.

In Table 2 below, a new user with user profile $\{0, 1, 1, 0, 1\}$ visits the page:

User Id				
Feature	$\underline{a}_1$	$\underline{a}_2$	$\underline{a}_3$	$\underline{u}$
cars	1	0	0	0
sports	1	1	1	1
cooking	0	1	1	1
Java	0	0	0	0
Internet	1	0	1	1

TABLE 2

The correlation is calculated by comparing the new user profile with each column of the web page profile using the similarity function across the web page profile.

$$\text{sim}(\underline{u}, \underline{a}_1) = .67$$

$$\text{sim}(\underline{u}, \underline{a}_2) = .81$$

$$\text{sim}(\underline{u}, \underline{a}_3) = 1.00$$

$$\text{correlation}(\underline{u}, \underline{A}) = (.67 + .81 + 1.00)/3 = .83$$

A match between a user and a page is determined by comparing the correlation against a threshold which may be pre-defined or optionally set by the user agent. For example, if the threshold $\theta = .80$, the user profile would be considered a match and

10

included in the page profile, because $.83 > .80$.

If the web page profile is particularly large, it may be necessary to employ some optimization techniques to speed the calculation. Many optimizations are possible, for example, summing over a random sample of columns of the page profile, rather than all.

The results of the match can be presented in several known ways, for example, through colour-coding the links or annotating the link with a numerical rating a confidence level, number or percentage in the recommendation (such as described, for example, Hill et al., "Recommending and Evaluating in a Virtual Community of Users, 1994, Bell Communications Research).

The search continues for each of the pages that meet the search criteria and has not been eliminated by testing for the threshold. The list of relevant pages is refined until they do not match the criteria, they contain no further links to other pages, or reach some pre-defined limit of number of results. Circular links can be handled by testing against a list of already successfully matched pages before a match is attempted.

The web pages which meet or exceed the threshold are presented to the user. In this manner, web pages, documents and information which meets both the search criteria of the user, as well as correlate with interests of similar users are delivered.

Turning to Figure 3, a preferred embodiment of the present invention, implemented in Java, is shown. The User Agent 302 operates as an applet stored on WWW Client 304. Profile Agent 1 (306) is connected to the same local area network 300 as the user Agent 302. Profile Agent 1 (302) is implemented as a Java application on WWW Server 308 and manages Page Profile database 310. Profile Agent 2 (312) and Profile Agent 3 (318) reside on remote WWW Servers 314 and 320 (www.soccer.com and www.cars.com), respectively. Both WWW Servers 314 and 320 are implemented supporting Java applications. WWW Client 304, Profile Agent 1 (302), Profile Agent 2 (306) and Profile Agent 3 (318) are provided with connections to the World Wide Web 324. WWW Client 304 is a standard Web browser such as Netscape Communicator or Microsoft Explorer and WWW Servers 306, 314 and 320 are standard Web servers such as Netscape FastTrack or Apache server.

Profile Agents 306, 312, and 318 implement the HTTP protocol and behave to WWW Client 304 just as a typical WWW server would. Each Profile Agent 306, 312, and 318 implements the following three commands: search, load, and ask. The search command is initiated by a user to conduct a search. The load command is used to
5 further refine a search. The ask command is used by profile agent 306, 312 and 318 to inquire about interest levels for linked pages. These commands have the following format:

```
search=uid:profile:keywords  
10 load=uid:profile:keywords  
ask=uid:profile
```

Each user is assigned a unique user id (uid) when the user is set-up with user agent 302. Each command also includes the profile of the user initiating the command.
15 The keywords which are part of the search and load commands are the words, separated by commas, that the user wishes to search. It is also possible to pass other information with each command, such as a threshold for assessing the relevance of a page as set by the user, as a further embodiment of the invention.

Turning to Figure 4, the processing of the commands is described in further
20 detail.

The User Agent 402 provides the mechanism for the user to define his user profile, as previously described with respect to Figure 1, or through the use of a separate combination box for selecting features.

User Agent 402 issues search command 404 on the initial screen 403 off which
25 the user starts his search. The user Agent 402 is embedded into the Web page for the initial screen 403 as a Java applet. This applet displays a typical search form in HTML format with a field for entering keywords 406 and button 408 to set off the search. The applet is downloaded when the initial screen 403 web page is retrieved from the WWW server from which the user starts his search. If the applet code is stored as
30 "SearchForm.class" on the WWW Server, the initial screen 403 HTML web page would have the following statement:

```
<applet code=SearchForm width=200 height=50>  
</applet>
```

5 The User Agent 402 on the initial screen 403 retrieves the user id and profile
410 from the WWW Client. The preferred embodiment utilizes the presence of a
“cookie” mechanism by which the web server connections (such as applets or CGI
scripts) can both store and retrieve information on the client side of the connection.
Such a mechanism is implemented by the major browsers, for example, Netscape
10 Communicator .

Cookies are stored as name/value pairs in a designated file on the WWW Client.
Each cookie is associated with a path and an optional expiry date. When the Java
15 applet requests a cookie from the WWW Client, the path component of the applet’s
document base is compared to the path attribute, and if there is a match, the cookie is
visible to the applet. Commercial browsers such as Netscape provide a Java class
library for accessing cookies from within an applet.

The cookie mechanism is applied as follows. When a first time user (which can
20 be indicated by passing a special uid with the search command) submits a search
through his User agent 402, the Profile Agent 412 at the server side generates a unique
user id and passes it back to the User Agent 402. The User Agent 402 then creates a
cookie on the client side of the connection that contains this user id and profile 410.
For example, the following cookie represents a user with user id “1234” and user
25 profile “01101” (which corresponds to the feature vector {0, 1, 1, 0, 1} using a
straightforward encoding):

```
uid=1234; profile=01101; path=/
```

30 When the User Agent 402 subsequently issues a search command 404 to the
Profile Agent 412 that created the cookie, it retrieves the user id and profile 410 from
the cookie and sends it to the Profile Agent 412. As discussed above, a complete search

command contains the user id, the user profile and the list of keywords. For example, the following is a search command to search the Web on behalf of a user with id "1234" and profile "01101" for the keyword "worldcup":

5 search=1234:01101:worldcup

Each WWW Server is set up having an index page which contains initial links from which to begin a search. This could be a database of keywords and associated web pages containing those key words. This database could be derived from a typical
10 search engine or robot. For example, if the user connects to a WWW Server to start his search, the index page might contain the following HTML initial links:

< a href=http://www.soccer.com/index.html > Soccer
< a href=http://www.cars.com/marketplace.html > Cars for sale
15 < a href=http://intranet/home.html > ACM

On receiving a search command, the Profile Agent 412 retrieves the index page from its WWW server, and issues an ask command 414 to one or more other Profile Agents 416 for recommendations on each document linked from the index page. With
20 each ask command the user id and profile as provided. For example:

ask=1234:01101

The Profile Agent 416 for the linked page replies with the level of interest
25 calculated in association with other profile agents 416 using a correlation function such as previously described for the given profile. Pages whose level of interest is above a certain pre-defined threshold, or a threshold optionally set up the user agent 402, are then downloaded and filtered against the optional list of keywords. In a simple filtering scheme, pages that do not include the keywords would be removed from the list of
30 recommended links. The Profile Agent 412 then modifies the links in the original page by encoding and including how interesting each of them would be to the user and sends the modified link page back to the User Agent 402. Each link is annotated, for

example, through color coding or a numerical indication of the degree of confidence that the page is relevant to the user.

For the above example, the Profile Agent 412 might have annotated the links as follows:

```
<a href=http://www.soccer.com/index.html?load=1234:01101:worldcup>
```

```
<font color="#FF000"> Soccer </a>
```

```
<a href=http://www.cars.com/marketplace.html>
```

```
10      Cars for sale </a>
```

```
<a href=http://intranet/home.html?load=1234:01101:worldcup>
```

```
<font color="#FF000"> Intranet homepage </a>
```

Here a color coding scheme with one threshold is used. Any link that was recommended with a correlation at or above threshold is encoded in red (corresponding to the color code FF000 in RGB format). Each color coded link also contains an embedded load command which contains the user id and profile 410 to be passed to the Profile Agent 412 for that page. The load command is separated from the actual link using a "?", which is a CGI (Common Gateway Interface) convention.

20 The search command is only invoked once. Subsequent refinements of the search are performed via load commands. For example:

```
load=1234:01101:worldcup
```

25 The processing of the load and search commands are generally the same, except for two aspects, which warrant the separation into two commands.

First, in the case of the load command, the host of the Profile Agent to receive indicated by the command is the host part of the URL which contains the load command. For example, if the user selects link

30 http://intranet/home.html?load=1234:01101:worldcup

in the page returned by Profile Agent 412, the following information will be sent (via the WWW client) in the load command 419 to the Profile Agent 420 on the host "intranet":

home.html?load=1234:01101:worldcup

The Profile Agent 420 then extracts from this a local path (home.html) and the
5 actual load command (load = 1234:01101:worldcup).

Second, the page returned by the Profile Agent 420 in reply to a load command
419 contains a Java applet 422 that monitors the time that the user spends reading the
page, which the Profile Agent 420 then uses as an indication of interestingness.

10

Further information on how recommendations are solicited from other Profile
Agents and how the level of interest displayed by a user in a particular page is measured
is described below.

On receiving a search command 404 or load command 419, a Profile Agent 412
15 or 420, as the case may be, first retrieves the appropriate page from the WWW Server
on the same site. This is either the index page or the page at the path which was passed
together with the search or load command. The Profile Agent 412 or 420, as the case
may be, then extracts links to other pages from the document. For each link the Profile
Agent 412 or 420 establishes a socket connection to the remote Profile Agent 416 or
20 426 using the URL for that link. This is not necessary if the page is already on the
local WWW Server and monitored by the same Profile Agent. The Profile Agent 412
or 420 then sends an ask command 414 or 424 to the Profile Agent or Agents 416 or
426 for the linked page and waits for it to reply with the interest level (correlation) for
that page. The Profile Agent 416 or 426 for the linked page computes the correlation
25 between the specified user profile 410 and the profiles stored for that page in the Page
Profile database and returns it to the Profile Agent 412 or 420 (as the case may be) as
the interest level. The socket is then deestablished, if necessary.

In the example of Figure 3, the User Agent 302 would first send a search
command with uid 1234, profile 01101, and keywords "worldcup" to Profile Agent
30 306, which would then load the page intranet/index.html for the WWW Server 308.
Profile Agent 1 (306) then sends ask commands with the same uid and profile to Profile
Agent 2 (312) on host www.soccer.com and Profile Agent 3 (318) on host

www.cars.com. From the replies Profile Agent 1 (306) would then assemble a modified index page with embedded load commands. If the user now selects the link "Soccer", WWW Client 304 would connect to Profile Agent 2 (312) which parses the path part of the URL into the name of a local page (index.html) to be retrieved from
 5 WWW Server www.soccer.com and a load command from Profile Agents linked from within the index.html page and sent a modified page back to the WWW Client 304.

To capture the time spent viewing a page, a User Agent applet that measures time is started on loading the page to WWW Client 304. When the user changes to a different page (by following a link, or using one of the browser buttons like "Back",
 10 "Forward" etc.), the User Agent 302 reports the time spent while the page was visible back to the Profile Agent 312 from which the page was loaded. The User Agent 302 applet is implemented as a invisible Java applet embedded into the Web page downloaded from the Profile Agent 312. To illustrate, assuming that code for the User Agent 302 is contained in the file "TimeTracker.class", the page assembled by the
 15 Profile Agent 312 must, for example, include the following statements:

```
<applet code=TimeTracker width=1 height=1 >
```

```
  <param name=uid value="1234" >
```

```
  <param name=profile value="01101" >
```

```
20 </applet>
```

The Profile Agent 312 at the server side records the information sent by the User Agent 302 which includes the user id, the user profile, the time spent reading the page and the URL of the page loaded. This information is then used to update the
 25 profile for that page in the Page Profile database 316. As described above, a page is considered interesting to the user if the time spent reading it is in a certain range. For such pages, the user profile is added to the page profile in the page profile database 316 at the position indicated by the user id. If an entry for the user previously existed in the profile it is overwritten. This procedural interaction between the user agent 302 and
 30 profile agent 312 is followed for all other profile agents interacting with the user agent during a search.

Although the invention has been described in terms of the preferred and several alternate embodiments, those skilled in the art will appreciate other modifications and alternation that can be made without departing from spirit and scope of the teachings of the invention. All such modifications are intended to be included within the scope of the

5 claims appended hereto.

CLAIMS:

1. A web search engine, comprising:

a user agent to interface with a user for defining a user profile, receiving at least one keyword input by a user for a search request, and initiating said search request having search parameters including said at least one keyword and said user profile;

a first profiling agent to receive said search request for initiating a search to identify candidate documents matching said search parameters;

a plurality of second profiling agents associated with said candidate documents, each of said candidate documents having an associated document profile for correlating said user profile to document profiles associated with said candidate documents to determine an interest level to each of said candidate documents;

wherein said user agent presents a listing of said candidate documents with interest levels to said user in response to said search; and

wherein each of said document profiles comprises user profiles of users who have retrieved and displayed on their viewing devices a candidate document associated with respective said document profile, for a time in a range indicating interest.

2. The web search engine of claim 1, wherein each of said second profiling agents further comprises updating said document profile associated with said candidate document with said user profile in response to receiving a communication from a program that was sent with said candidate document to said user which indicates that said user has retrieved and displayed said candidate document for said time in a range indicating interest.

3. The web search engine of claim 2, wherein said user profiles and said document profiles are managed by agents.

4. The web search engine of claim 2, wherein said program is an applet.

5. The web search engine of claim 1, wherein said listing of each of said candidate documents is presented in response to said interest level being above a threshold.

6. The web search engine of claim 1, wherein said interest level is calculated by correlation formula, as follows:

$$\sum_{M=1}^{k=l} \text{sim}(\underline{u}, \underline{a}_k) / M, \text{ where:}$$

- (a) $\underline{u}$ is said user profile;
- (b) $\underline{a}_k$ is k^{th} column of said document profile.
- (c) M is number of said user profiles in said document profile; and
- (d) $\text{sim}(\underline{u}, \underline{a}_k)$ is the similarity between said user profile and the k^{th} column of said document profile.

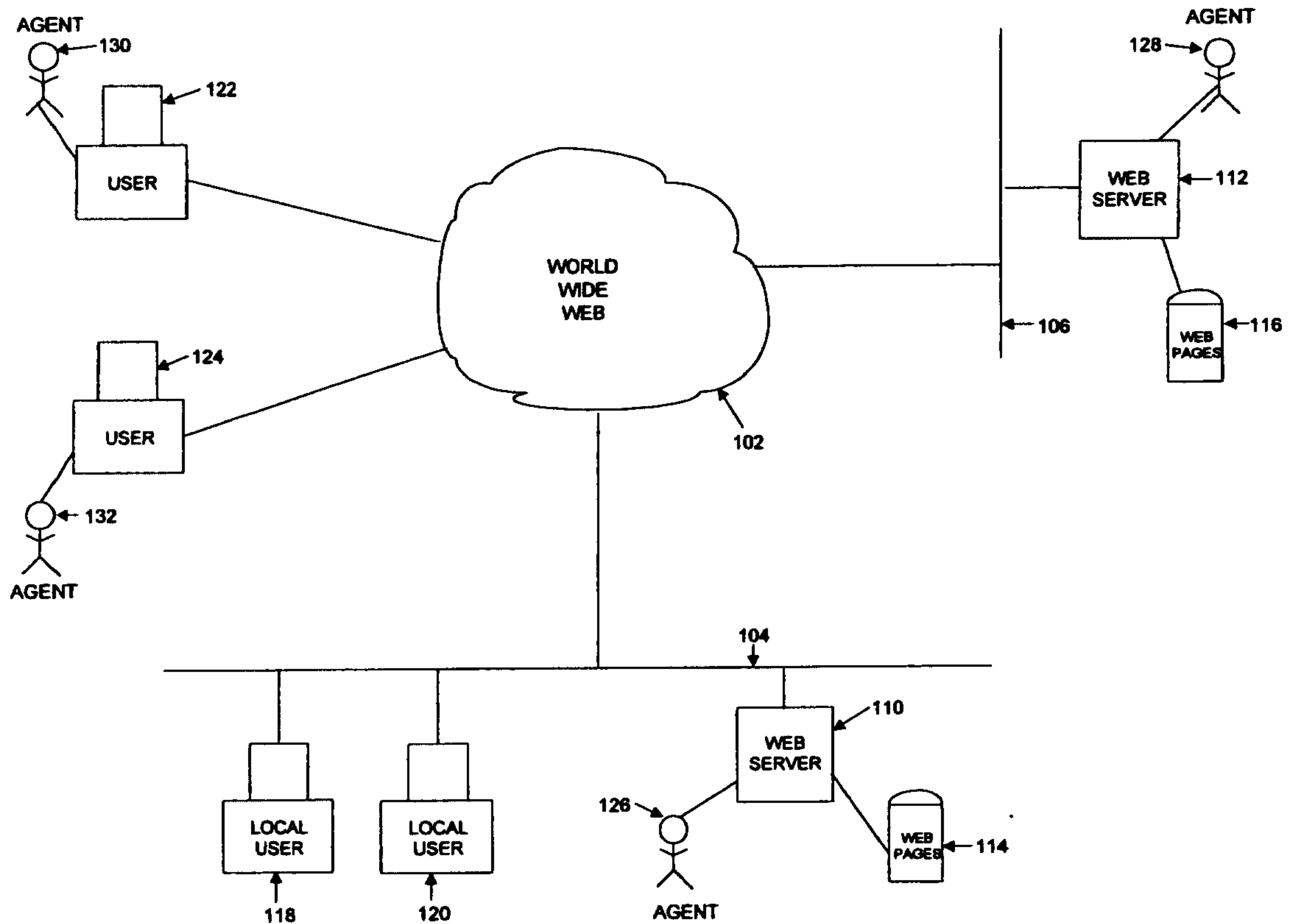


Figure 1

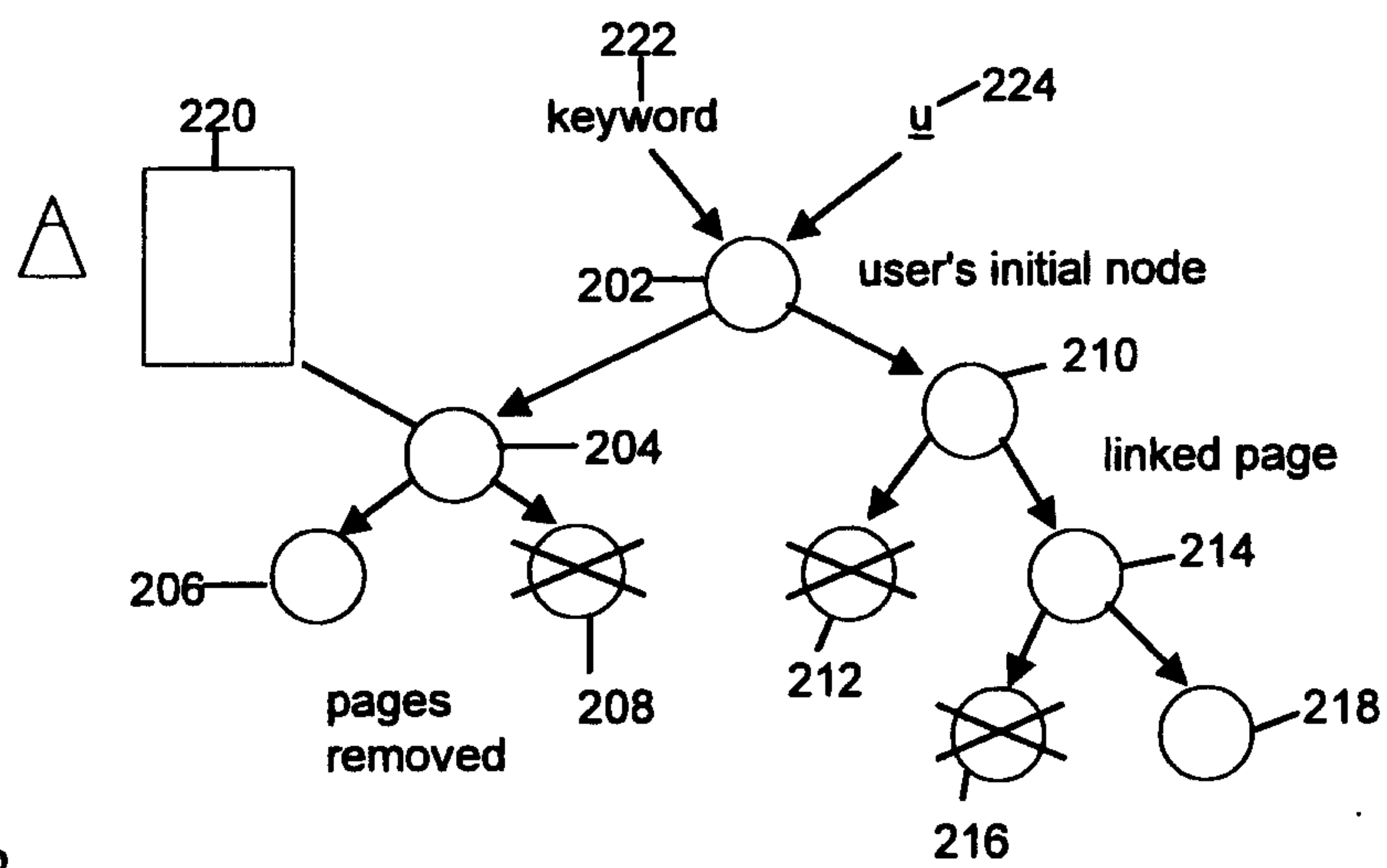


Figure 2

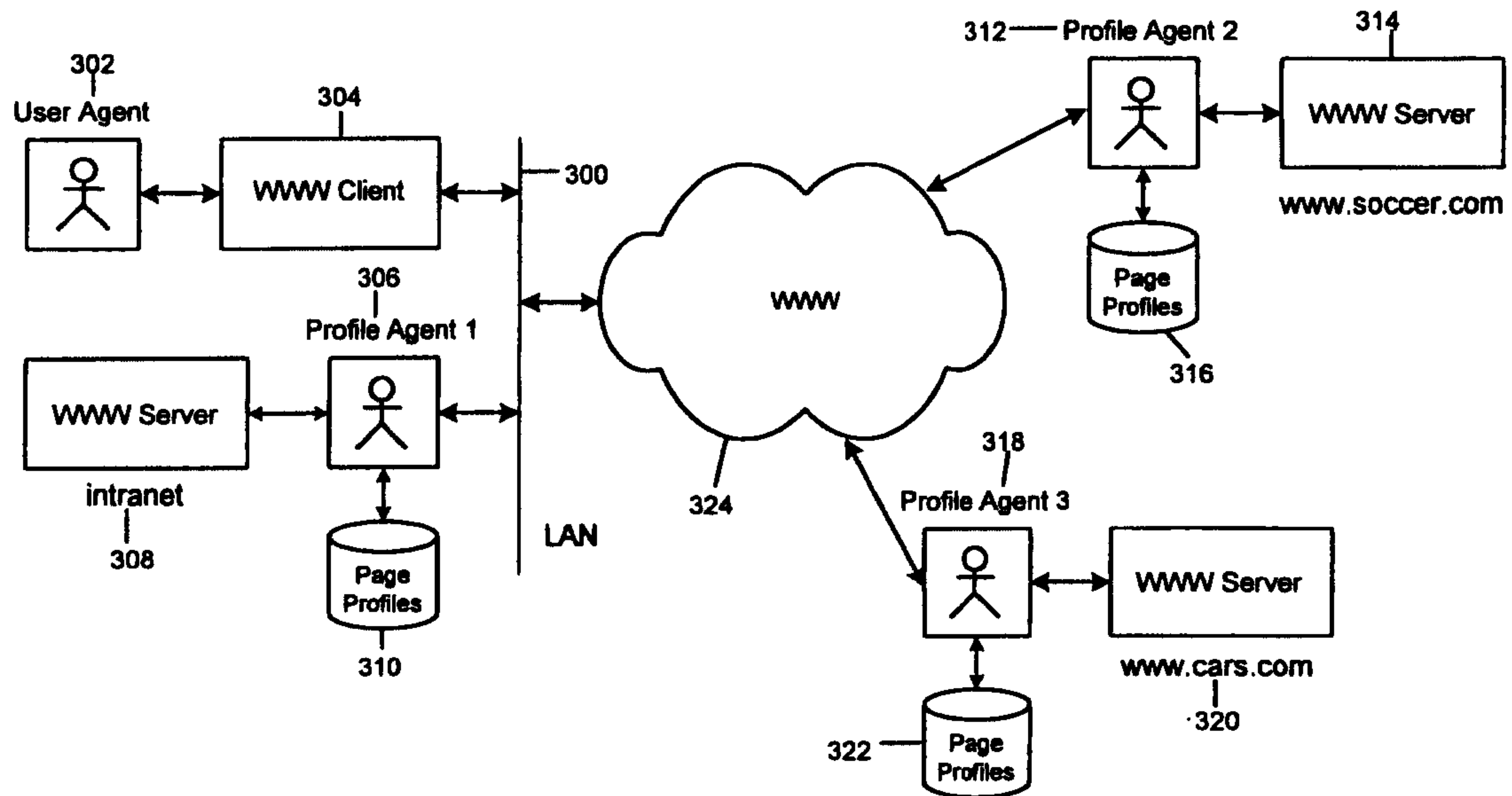


Figure 3

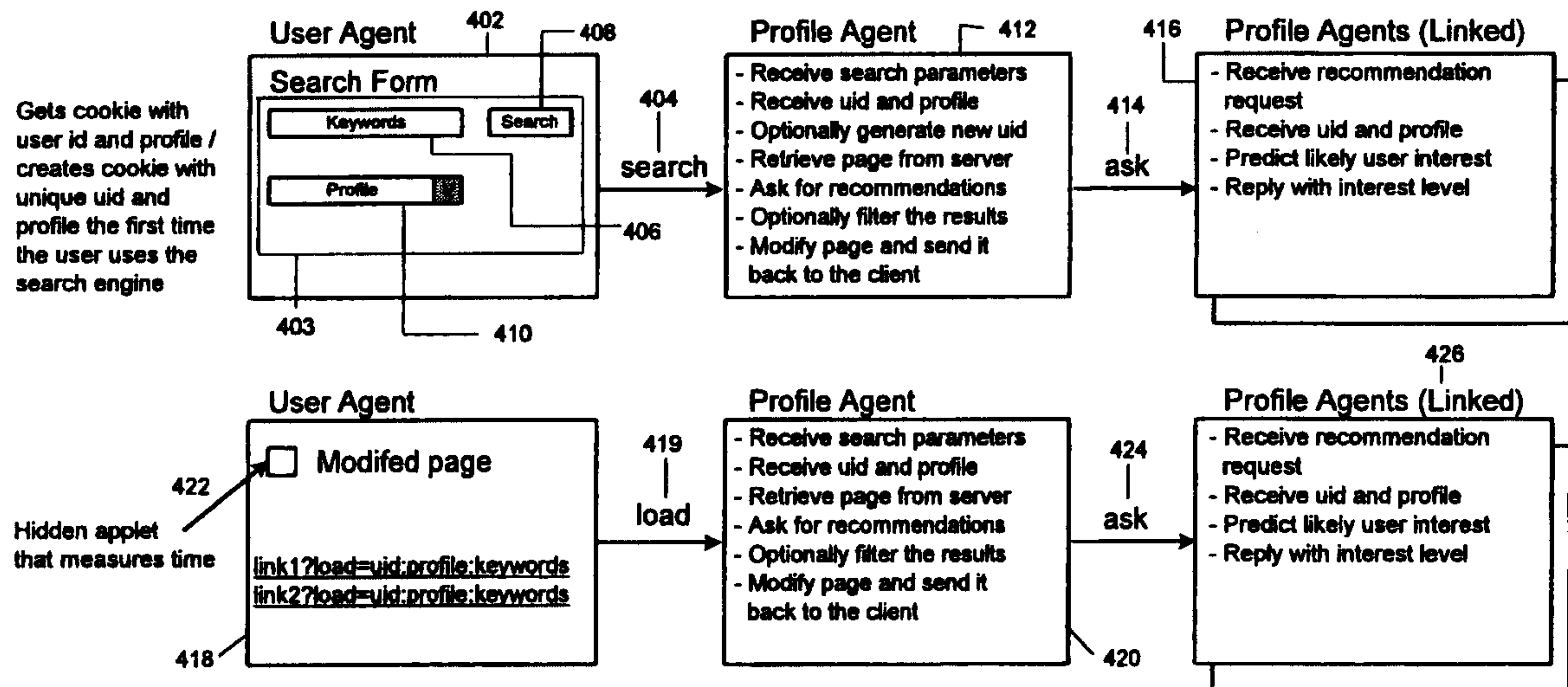
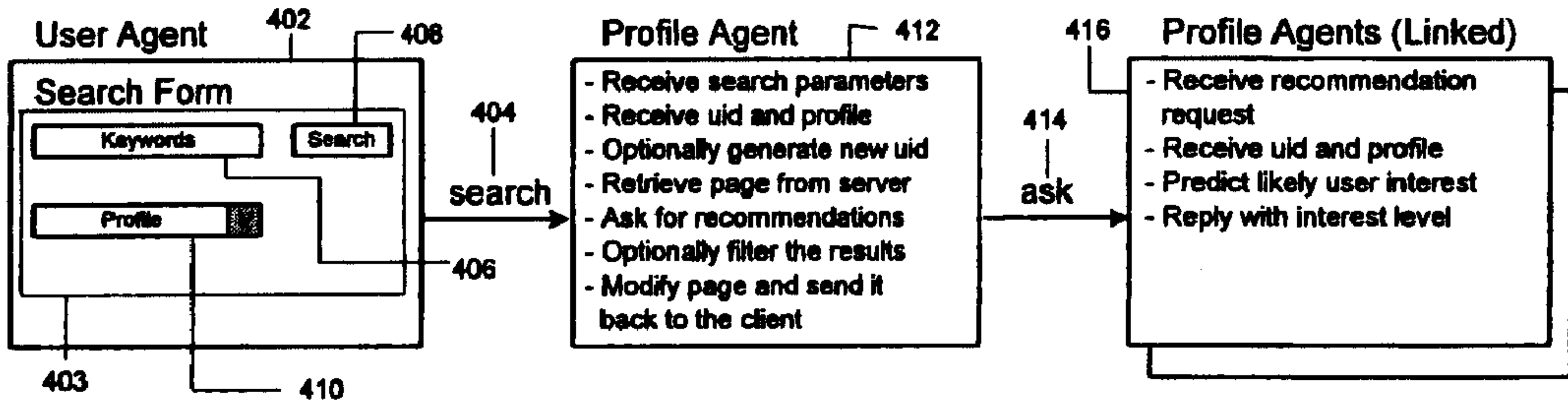


Figure 4

Gets cookie with user id and profile / creates cookie with unique uid and profile the first time the user uses the search engine



Hidden applet that measures time

