2

User Profiles for Personalized Information Access

Susan Gauch[1], Mirco Speretta[1], Aravind Chandramouli[1] and Alessandro Micarelli[2]

[1] Electrical Engineering and Computer Science
Information & Telecommunication Technology Center
2335 Irving Hill Road, Lawrence Kansas 66045-7612
{sgauch, mirco, aravindc}@ittc.ku.edu
[2] Department of Computer Science and Automation
Artificial Intelligence Laboratory
Roma Tre University,
Via della Vasca Navale, 79 00146 Rome, Italy
micarel@dia.uniroma3.it

Abstract. The amount of information available online is increasing exponentially. While this information is a valuable resource, its sheer volume limits its value. Many research projects and companies are exploring the use of personalized applications that manage this deluge by tailoring the information presented to individual users. These applications all need to gather, and exploit, some information about individuals in order to be effective. This area is broadly called user profiling. This chapter surveys some of the most popular techniques for collecting information about users, representing, and building user profiles. In particular, explicit information techniques are contrasted with implicitly collected user information using browser caches, proxy servers, browser agents, desktop agents, and search logs. We discuss in detail user profiles represented as weighted keywords, semantic networks, and weighted concepts. We review how each of these profiles is constructed and give examples of projects that employ each of these techniques. Finally, a brief discussion of the importance of privacy protection in profiling is presented.

2.1 Introduction

In the modern Web, as the amount of information available causes information overloading, the demand for personalized approaches for information access increases. Personalized systems address the overload problem by building, managing, and representing information customized for individual users. This customization may take the form of filtering out irrelevant information and/or identifying additional information of likely interest for the user. Research into personalization is ongoing in the fields of information retrieval, artificial intelligence, and data mining, among others.

This chapter discusses user profiles specifically designed for providing personalized information access. Other types of profiles, build using different construction techniques, are described elsewhere in this book. In particular, Chapter 4 [40] discusses generic user modeling systems that are broader in scope, not necessarily focused on Internet applications. Related research on collaborative recommender systems, discussed in Chapter 9 of this book [81], combines information from multiple users in order to provide improved information services. Concern over privacy protection is growing in parallel with the demand for personalized features. These two trends seem to be in direct opposition to each other, so privacy protection must be a crucial component of every personalization system. A detailed discussion can be found in Chapter 21 of this book [39].

There are a wide variety of applications to which personalization can be applied and a wide variety of different devices available on which to deliver the personalized information. Early personalization research focused on personalized filtering and/or rating systems for e-mail [49], electronic newspapers [14, 16], Usenet newsgroups [41, 58, 86, 91, 106], and Web documents [4]. More recently, personalization efforts have focused on improving navigation effectiveness by providing browsing assistants [9, 13], and adaptive Web sites [69]. Because search is one of the most common activities performed today, many projects are now focusing on personalized Web search [46, 88, 92] and more details on the subject can be found in Chapter 6 of this book [52]. However, personalized approaches to searching other types of collections, e.g., short stories [76], Java source code [100], and images [14] have also been explored. Commercial products are also adopting personalized features, for example, Yahoo!'s personalized Web portals [110] and Google Lab's personalized search [30].

The aforementioned systems are just a few examples that illustrate the breadth of applications to which personalized approaches are being investigated. Nichols [63] and Oard and Marchionini [64] provide a general overview of some the issues and approaches to personalized rating and filtering and Pretschner [71] describes approximately 45 personalization systems.

Most personalization systems are based on some type of user profile, a data instance of a user model that is applied to adaptive interactive systems. User profiles may include demographic information, e.g., name, age, country, education level, etc, and may also represent the interests or preferences of either a group of users or a single person. Personalization of Web portals, for example, may focus on individual users, for example, displaying news about specifically chosen topics or the market summary of specifically selected stocks, or a groups of users for whom distinctive characteristics where identified, for example, displaying targeted advertising on e-commerce sites.

In order to construct an individual user's profile, information may be collected *explicitly*, through direct user intervention, or *implicitly*, through agents that monitor user activity. Although profiles are typically built only from topics of interest to the user, some projects have explored including information about non-relevant topics in the profile [35, 104]. In these approaches, the system is able to use both kinds of topics to identify relevant documents and discard non-relevant documents at the same time.

Profiles that can be modified or augmented are considered *dynamic*, in contrast to *static* profiles that maintain the same information over time. Dynamic profiles that

take time into consideration may differentiate between short-term and long-term interests [37, 93, 103]. *Short-term* profiles represent the user's current interests whereas *long-term* profiles indicate interests that are not subject to frequent changes over time. For example, consider a musician who uses the Web for her daily research. One day, she decides to go on vacation, and she uses the Web to look for hotels, airplane tickets, etc. Her user profile should reflect her music interests as long-term interests, and the vacation-related interests as short-term ones. Once the user returns from her vacation, she will resume her music-related research, and the vacation information in her profile should eventually be forgotten. Because they can change quickly as users change tasks, and less information is collected, short-term user's interests are generally harder to identify and manage than long-term interests. In general, the goal of user profiling is to collect information about the subjects in which a user is interested, and the length of time over which they have exhibited this interest, in order to improve the quality of information access and infer user's intentions.

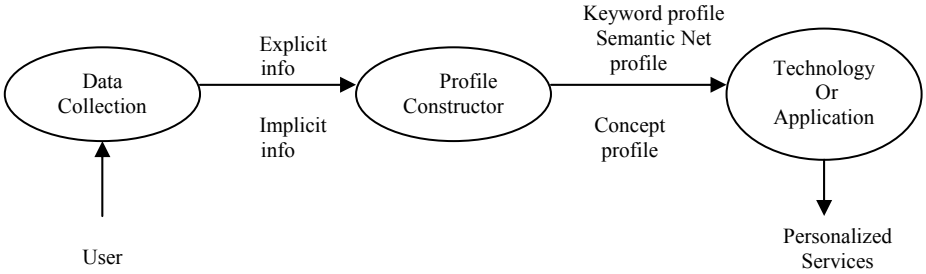

Fig. 2.1. Overview of user-profile-based personalization

As shown in Figure 2.1, the user profiling process generally consists of three main phases. First, an information collection process is used to gather raw information about the user. As described in Section 2.2, depending on the information collection process selected, different types of user data can be extracted. The second phase focuses on user profile construction from the user data. Section 2.3 summarizes a variety of ways in which profiles may be represented and Section 2.4 some of the ways a profile may be constructed. The final phase, in which a technology or application exploits information in the user profile in order to provide personalized services, is discussed in Parts II and III of this book.

2.2 Collecting Information About Users

The first phase of a profiling technique collects information about individual users. A basic requirement of such a system is that it must be able to uniquely identify users. This task is described in more detail in Section 2.2.1. The information collected may be explicitly input by the user or implicitly gathered by a software agent. It may be collected on the user's client machine or gathered by the application server itself. Depending on how the information is collected, different data about the users may be extracted. Several options, and their impacts, are discussed in Section 2.2.2. In

general, systems that collect implicit information place little or no burden on the user are more likely to be used and, in practice, perform as well or better than those that require specific software to be installed and/or explicit feedback to be collected.

2.2.1 Methods for user identification

Although accurate user identification is not a critical issue for systems that construct profiles representing groups of users, it is a crucial ability for any system that constructs profiles that represent individual users. There are five basic approaches to user identification: software agents, logins, enhanced proxy servers, cookies, and session ids. Because they are transparent to the user, and provide cross-session tracking, cookies are widely used and effective. Of these techniques, cookies are the least invasive, requiring no actions on the parts of users. Therefore, these are the easiest and most widely employed. Better accuracy and consistency can be obtained with a login-based system to track users across sessions and between computers, if users can be convinced to register with the system and login each time they visit. A good compromise is to use cookies for current sessions and provide optional logins for users who choose to register with a site.

Web usage mining can also be used to identify users, and these approaches are covered in more detail in Chapter 3 of this book [59]. Many companies rely on data aggregators, such as Acxiom [1], to provide demographic data about customers. This information actually turns out to be more accurate than surveys of customers themselves. Usually, all that is required to get full demographic data is a credit card number or the combination of name and zipcode, information that is often collected during purchase or registration.

The first three techniques are more accurate, but they also require the active participation of the user. Software agents are small programs that reside on the user's computer, collecting their information and sharing this with a server via some protocol. This approach is the most reliable because there is more control over the implementation of the application and the protocol used for identification. However, it requires user-participation in order to install the desktop software. The next most reliable method is based on logins. Because the users identify themselves during login, the identification is generally accurate, and the user can use the same profile from a variety of physical locations. On the other hand, the user must create an account via a registration process, and login and logout each time they visit the site, placing a burden on the user. Enhanced proxy servers can also provide reasonably accurate user identification. However, they have several drawbacks. They require that the user register their computer with a proxy server. Thus, they are generally able to identify users connecting from only one location, unless users bother to register all of the computers they use with the same proxy server.

The final two techniques covered, cookies and session ids, are less invasive methods. The first time that a browser client connects to the system, a new userid is created. This id is stored in a cookie on the user's computer. When they revisit the same site from the same computer, the same userid is used. This places no burden on the user at all. However, if the user uses more than one computer, each location will have a separate cookie, and thus a separate user profile. Also, if the computer is used

by more than one user, and all users share the same local user id, they will all share the same, inaccurate profile. Finally, if the user clears their cookies, they will lose their profile altogether, and if users have cookies turned off on their computer, identification and tracking is not possible. Session ids are similar, but there is no storage of the userid between visits – each user begins each session with a blank slate, but their activity during the visit is tracked. In this case, no permanent user profile can be built, but adaptation is possible during the session.

2.2.2 Methods for user information collection

User profiles may be based on heterogeneous information associated with an individual user or a group of users who showed similar interests or similar navigational behavior. Broadly, user profile construction techniques can be partitioned by the type of input used to build the profile. In this section we discuss explicit and implicit feedback systems in detail. Hybrid approaches are also possible. Papazoglou [66] uses an automatic component to build a user profile based on user observations, but they also provide a mechanism for explicit relevance feedback in order to better tailor the profiles to user's individual interests.

Explicit user information collection Explicit user information collection methodologies, often called explicit user feedback, rely on personal information input by the users, typically via HTML forms. The data collected may contain demographic information such as birthday, marriage status, job, or personal interests. In addition to simple checkboxes and text fields, a common feedback technique is the one that allows users to express their opinions by selecting a value from a range. All these methodologies have the drawback that they cost the user's time and require the user's willingness to participate. If users do not voluntarily provide personal information, no profile can be built for them.

Commercial systems have been exploring customization for some time. Many sites collect user preferences in order to customize interfaces. This customization can be viewed as the first step to provide personalized services on the Web. Many of the systems described in Section 2.4 rely on explicit user information. The collection of preferences for each user can be seen as a user profile and the services provided by these applications adapt in order to improve information accessibility. For instance, MyYahoo! [110], explicitly ask the user to provide personal information that is stored to create a profile. The Web site content is then automatically organized based on the user's preferences.

More sophisticated personalization projects based on explicit feedback have focused on navigation. One of the earliest, Syskill & Webert [68], recommends interesting Web pages based on explicit feedback. If the user rates some links on a page, Syskill & Webert can recommend other links on the page in which they might be interested. In addition, the system can construct a Lycos query and retrieve pages that might match a user's interest. The Wisconsin Adaptive Web Assistant (WAWA) [84,85] also uses explicit user feedback to train neural networks to assist users during browsing.

One problem with explicit feedback is that it places an additional burden on the user. Because of this, or privacy concerns, the user may not choose to participate. Users may not accurately report their own interests or demographic data, or, since the profile remains static whereas the user's interests may change over time, the profile may become increasingly inaccurate over time. An argument in favor of explicit feedback is that, in some cases, users enjoy providing, and sharing, their feedback. This is most evident in movie rating sites such as NetFlix [62] and sites dedicated to collecting, and sharing, consumer ratings such as ePinions [24].

Implicit user information collection User profiles are often constructed based on implicitly collected information, often called implicit user feedback. The main advantage of this technique is that it does not require any additional intervention by the user during the process of constructing profiles. Kelly and Teevan [36] give an overview of the most popular techniques used to collect implicit feedback, and the type of information about the user that can be inferred from the user's behavior. Table 2.1 summarizes the approaches covered in this chapter, the type of information each approach is able to collect, and the breadth of applicability of the collected information. Because they only require a one time setup, do not require new software to be developed and installed on the user's desktop, and only track browsing activity, proxy servers seem to be a good compromise between easily capturing information and yet not placing a large burden on the user. Capturing activity at the site providing personalized services, for example a search site itself, is also an option in some cases. It requires absolutely no special user activity, but not all personalized sites are used frequently enough by any single user to allow them to create a useful profile.

Table 2.1. Implicit User Information Collection Techniques

Collection Technique	Information Collected	Information Breadth	Pros and Cons	Examples
Browser Cache	Browsing history	Any Web site	**pro**: User need not install anything. **con:** User must upload cache periodically.	OBIWAN [71]
Proxy Servers	Browsing activity	Any Web site	**pro**: User can use regular browser. **con:** User must use proxy	OBIWAN [71] Trajkova [99] Barrett et al [6]

			server.	
Browser Agents	Browsing activity	Any personalized application	**pro**: Agent can collect all Web activity. **con:** Install software and use new application while browsing.	Letizia [43] WebMate [13] Vistabar [50] WebWatcher [58]
Desktop Agents	All user activity	Any personalized application	**pro**: All user files and activity available. **con:** Requires user to install software.	Seruku [83] Surfsaver [94] Haystack [2,17] Google Desktop [29] Stuff I've Seen [22]
Web Logs	Browsing activity	Logged Web site	**pro**: Information about multiple users collected. **con:** May be very little information since only from one site.	Mobasher [59]
Search Logs	Search	Search engine site	**pro**: Collection and use of information all at same site. **con:** Cookies must be turned on and/or login to site. **con:** May	Misearch [87] Liu et al [45]

			be very little information	

Browsing histories are a common source of information from which user interests are extracted. Browsing histories are collected in two main ways: users share their browsing caches on a periodic basis [71]; or users install a proxy server that acts as their gateway to the Internet, thereby capturing all Internet traffic generated by the user [6, 99]. These browsing histories contain the urls visited by the user and the dates and times of the visits. Summary information about the number of visits to a particular url over a variety of time periods can be easily extracted. The time spent on the each page can also be inferred, with some error, as the time between consecutive hyperlink clicks. These browsing histories are typically shared with one particular Web site, allowing that site only to provide personalized services. Another drawback to this approach is that it typically only collects the user's browsing history from a single computer. However, a user could share their browsing caches from multiple computers or install the same proxy server on each computer they use regularly (e.g., home and work). Even if they do not do this, they could, via a login system, use the same user profile in multiple locations, allowing consistent access to personalized services.

Many personalization approaches use agents to collect information interactively, while the user browses. These browser agents are implemented as either a stand-alone application that includes browsing capabilities or a plug-in to an existing browser. Because the browser agents are installed on the user's desktop computer, they are able to capture all of the activities the user performs while browsing. Although not every system collects or uses all available information, this approach allows the system to collect a richer set of information about the user than is available via browsing histories. In addition to the urls visited and accurate information about the amount of time spent on each Web page, the agents can also collect actions performed on the Web page such as bookmarking and downloading to disk. Letizia [43, 44] was one of the first systems to interactively collect and exploit implicit user feedback. Based on previously visited pages and bookmarked pages, it suggests links on the current page that might be of interest. Other browsing assistants based on browsing agents are WebMate [13], Vistabar [50], and Personal WebWatcher [58]. Some literature in this area distinguishes between browsing assistants and browsing agents. Vistabar [50] is a prototypical browsing assistant, a tool that helps users track viewed urls, fill out forms or fetch pages without any specific agenda. In contrast, WebMate [13] and Personal WebWatcher [58] are examples of browsing agents that perform more critical tasks such as highlighting hyperlinks of likely interest to the user, recommending urls, or refining search keywords.

One drawback to this approach is that it requires the user to install a new application on their computer and, in the case of a stand-alone browsing application, it requires them to use a new application during browsing instead of a conventional browser. Another drawback is that this approach requires a large investment in software development and maintenance. In order to capture user information, the personalization system must develop a high quality browsing agent or plug-in, distribute it widely, and maintain and support numerous, widely-deployed versions

that would result should the personalized application become successful. A final drawback to this approach is that, since it is resident on a personal computer, the user profile built would typically only be available when the user was using that particular computer. However, this drawback may be offset by the fact that, since it is resident on the user's computer, the user profile could be shared by multiple personalized applications.

There has been a recent surge in the availability of commercial toolbars and browser add-ons that include personalized features. Examples include the Seruku Toolbar [83] and SurfSaver [94], both of which try to help users organize their browsing histories stored in their desktop caches. These products are the direct descendents of the early browser agents developed by the research projects described above. Eventually, these personalized agents may evolve into a fully integrated personalized environment. In such a system, the searches would not be limited to the Web, but they would also include databases to which the user has access, and the user's personal documents. Such search systems are implemented in tools like Google Desktop Search [29] and Stuff I've Seen [22]. Then, the information found in the personal documents and databases could be used to enhance the user profile. The Haystack project [2, 17] presents the infrastructure necessary to create a personalized environment: a general purpose database to store all of the user's documents, the database management system, and the learning module in charge of maintaining the user profiles.

The above approaches all focus on collecting information about the users as they browse or perform other activities. Because they try to capture and share what the user is doing on their computer, they are essentially client-side approaches. All client-side approaches place some burden on the users in order to collect and/or share the log of their activities. In contrast, the final two approaches collect only the activities the user performs while interacting with the site providing the personalized services. Although they have access to less information than client-side approaches, they place no burden on the user at all, and can silently collect the information via cookies, logins, and/or session ids. There are two main sources of information for server-side personalization, browsing activity on the site and search interactions. Web logs capture the browsing histories for individual users at a given website. This information can be used to create Web sites that adapt their organization based on the user's behavior. Since web log mining is covered in detail in Chapter 3 of this book [59], it will not be discussed further here. However, search histories are discussed in some detail below.

Recently, search histories have been explored as a source of information for user profiling that can then be exploited to provide personalized search. Search histories contain information about the queries submitted by a particular user and the dates and times of those queries. The personalization system can also cache the urls and snippets of the result sets for each user's queries simultaneously with formatting that information for presentation to the user. If the personalization system wraps the presented results appropriately, the user clicks on particular results can also be collected. The personalization system could also download the complete Web pages for the visited urls. However, the network delays for this process are such that this cannot be done quickly enough to provide acceptable interactivity. Although downloading could be done as an offline process, this source of information is rarely

used. As mentioned previously, this approach has the advantage that user does not need to install a desktop application or plug-in to collect their activities and/or upload their information to the personalized service. The service that is providing the personalized search collects the user activities as the user interacts directly with the site. If the site requires a login process, the same profile can be used whenever they visit the site regardless of the particular computer they are using. The disadvantage is that because only the activities at the search site itself are tracked, much less information is available. Also, the amount of representative text collected per interaction, i.e., the queries and/or snippets, is much less than the full text of Web pages typically collected for browsing-based profiles. However, several projects [45, 88] have been able to successfully provide personalized search by building user profiles based on this information.

Comparing implicit and explicit user information collection Only recently have researchers begun to investigate the most effective source of information on which to build profiles. In 2000, Quiroga and Mostafa [73] compared systems using explicit feedback, implicit feedback, and a combination of the two by studying 18 users searching a collection of 6,000 health records classified into 15 different topics. Each user used the system for 15 sessions, and the highest precision of approximately 68% was achieved with profiles build from combined feedback. In contrast, explicit feedback alone produced a maximum precision of around 63% and the implicit feedback alone produced a maximum precision of around 58%. These differences were found to be statistically significant, suggesting that systems using the explicitly created profile or a profile built from a combination of explicit and implicit feedback produced better results than a system that made use of an implicitly created profile alone.

However, in contrast to the above findings, White et al. [102] did not find significant differences between profiles constructed using implicit and explicit feedback. They developed a system that used both implicit and explicit feedback to improve search on the Web. To compare these systems, they performed experiments with 16 users who searched the Web to answer specific questions on four topics. The successful completion of the task, the amount of time, and the number of result pages viewed to perform the task were used as metrics to evaluate the systems. The users who used the implicitly constructed profile were able to complete 61 out of the 64 tasks, while the users who used the explicitly created profile were able to complete only 57 tasks. Also, the average time per task for users with the implicit profile was 372 seconds, while the users with the explicit profile spent on 437 seconds on average. However, users with implicitly created profiles viewed approximately 3.3 results pages per task, more than the 2.5 pages viewed by users with explicitly created profiles. Since none of these differences was statistically significant, the authors concluded that implicit and explicit feedback were somewhat interchangeable.

In 2004 Wærn [100] studied the effect of user intervention on automatic filtering. The author compared the effectiveness of user profiles that were partially or completely built with automatic means. The study showed that although user intervention during profile construction can be useful, were not able to judge the quality of filtering and, furthermore, they were not able to improve the filters that were performing adequately.

Most recently, in 2005, Teevan et al. [98] evaluated a variety of information sources available to a client-side profiling agent, i.e., the Web pages visited, emails exchanged, calendar items, and all other documents stored on the client machine. Different rules, generating different collections, were used to gather information about the user, for example, recent documents only, Web pages only, documents only, and combinations of sources. In addition, two "lighter-weight" profiles were created: one constructed from search histories (queries issued in the past) and another from a list of all domains visited while browsing. They found that the richer the amount of information available, the better the profile performed. In particular, they found that the user profile built from the user's entire desktop index (the set of all information created, copied, or viewed by the user) was most accurate, followed by the profile built from recent information only, then that based on Web pages only. The least accurate profile was built from user-submitted queries only, but even it outperformed non-personalized search. They were also able to show that the profiles built from text collected implicitly from the user's desktop index could perform better than profiles built from explicit relevance feedback, a very promising result for future personalization systems.

These three studies, taken together, show that there is no clear answer on whether implicitly created profiles are more or less accurate than explicitly created profiles. However, the trend seems to be that the earlier study found explicit feedback better, the next study that the two forms of feedback were comparable, and the most recent study that implicit feedback was superior. This may indicate that, as experience with ways to collect and use implicit feedback has grown, the quality of the profiles constructed from this type of information improved. Since implicit feedback places less burden on the user, and it automatically updates as the user interacts with the system, it seems to be the preferable method of collecting information about users. One drawback to implicit feedback techniques is that they can typically only capture positive feedback. When a user clicks on an item or views a page, it seems reasonable to assume that this indicates some user interest in the item. However, it is not as clear, when a user fails to examine some data item, that this is an indication of disinterest. Thus, in general, implicit feedback techniques do not collect negative feedback.

2.3 User Profile Representations

User profiles are generally represented as sets of weighted keywords, semantic networks, or weighted concepts, or association rules. Because association rules are primarily used in the field of Web log mining, the subject of Chapter 3 of this book [59], they will not be discussed further here. Keyword profiles are the simplest to build, but because they fundamentally have to capture and represent all (or most) words by which interests may be discussed in future documents, they require a large amount of user feedback in order to learn the terminology by which a topic might be discussed. This problem is also shared by most semantic network-based profiles – they must learn the terminology with which concepts are discussed. Concept profiles, in contrast, are trained on examples for each concept *a priori*, and thus begin with an

existing mapping between vocabulary and concepts. Thus, they can build profiles that are robust to variations in terminology with less user feedback. Many of the approaches described in this section rely on extracting, and weighting, keywords from documents and comparing documents to each other. The reader is referred to Chapter 5 of this book [54] on document representations for discussions of term weighting, vector representations of documents, and document similarity calculations.

2.3.1 Keyword Profiles

The most common representation for user profiles is sets of keywords. These can be automatically extracted from Web documents or directly provided by the user. Weights, which are usually associated with keywords, are numerical representations of user's interests. Each keyword can represent a topic of interest or keywords can be grouped in categories to reflect a more standard representation of user's interests. An example of a weighted keyword-based user profile is shown below in Figure 2.2.

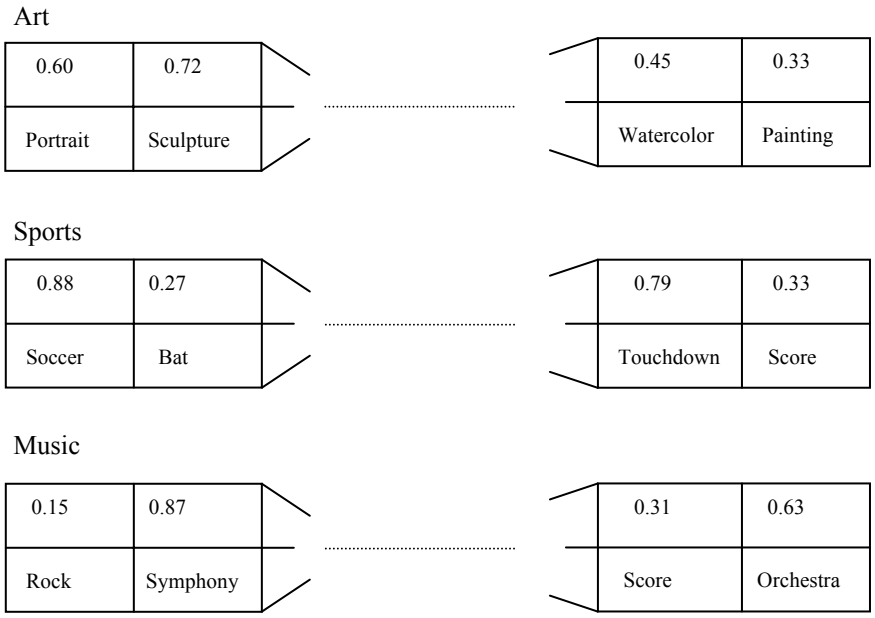

Fig. 2.2. A keyword-based user profile

Profiles represented in this way were among the first to be explored. The keywords in the profile are extracted from documents visited by the user during browsing, Web pages bookmarked or saved by the user, or the keywords were explicitly provided by the user. Each keyword is usually associated with a numerical

weight representing its importance in the profile. *Amalthaea* [61] is one of many systems that creates keyword profiles by extracting keywords from Web pages. They weight the keywords with the widely used tf*idf weighting scheme from information retrieval [80]. Each profile is represented in the form of a keyword vector, and the documents that are retrieved by the system in response to a search are converted to similar weighted keyword vector. These vectors are then compared to the profile using the cosine formula [80], and only the corresponding documents for those vectors that are closest to the profile are then passed on to the user. The system also provides the user with the option of explicitly specifying their profile, which is weighted higher than the profile built by the system. This project is somewhat unique in that it employs a learning algorithm based on genetic algorithms to adapt and expand the user profiles. Weighted keyword vectors have also been used in *Anatagonomy* [78], a personalized online newspaper, *Fab* [5], a Web page recommender, and *Letizia* [43], a browsing assistant, and *Syskill & Webert* [68] a recommender system.

PEA [60] is also a personalized Web search assistant that builds keyword-based profiles using terms extracted from the user's bookmarked Web pages. However, it differs from the other approaches in that, rather than creating a single profile for the user, the user is represented as a set of keyword/weight vectors, one per bookmark. The rationale behind this extension is that, if a user is interested in two topics, combining the keywords from both topics in a single vector results in a profile that points halfway between them. In contrast, representing each area of interest, as indicated by a bookmark, as a separate vector is likely to provide a more accurate profile. As the user browses, additional pages are recommended to user when the vector for a potential new page is similar to a vector for an existing bookmark. *WebMate* [13] also builds user profiles containing one keyword vector per user's area of interest whereas *Alipes* [103] expands upon this approach by representing each interest with three keyword vectors, i.e., a long-term descriptor and two short-term descriptors, one positive and one negative.

PSUN [91], a personalized system for reading Usenet news, improves on the keyword vector representation by representing user profiles using weighted word sequences. The profiles are thus made up of weighted n-grams, i.e., word sequences of length n. Each n-gram has an associated weight that estimates the likelihood of the words in the n-gram co-occurring in a document and a strength that represents the importance of that n-gram relative to all other n-grams in the profile. One of the main drawbacks to keyword-based profiles is that many words have multiple meanings. Because of this polysemy, the keywords in the user profile are ambiguous, making the profile inaccurate. By focusing on word sequences, which are essentially statistically derived phrases, the contexts of the individual words are constrained. They report that profiles built from n-grams of length 2, i.e., word pairs, are more accurate than profiles built from individual keywords, but no formal analysis is presented.

More details about personalization based on keyword profiles can be found in Chapter 10 of this book [67].

2.3.2 Semantic Network Profiles

In order to address the polysemy problem inherent with keyword-based profiles, the profiles may be represented by a weighted semantic network in which each node represents a concept. Minio and Tasso [56] explore an approach based on this in which each node contains a particular word found in the corpus and arcs are created based upon co-occurrences of the two words in the connected nodes. Their user model is further enhanced by the inclusion of a set of attribute-value pairs corresponding to the structured part of the documents, e.g., host, size, number of images, etc., that have previously been of interest to the user [4]. The SiteIF project also uses a word-based semantic to represent user profiles [92]. However, they found that representing individual words as nodes in the semantic network was not accurate enough to discriminate word meanings. Instead, they used information inherent in WordNet to group related words together in concepts called "synonym sets," or synsets. They represent a user profile as a semantic network in which the nodes are synsets, the arcs are co-occurrences of the synset members within a document of interest to the user, and the node and arc weights represent the user's level of interest.

InfoWeb [28], a filtering system for online digital libraries documents, also builds semantic network based profiles that represent long-term user interests. Each user profile is represented as a semantic network of concepts. Initially, each semantic network contains a collection of unlinked nodes in which each node represents a concept. Concept nodes, called *planets*, contain a single, representative weighted term for that concept. As more information about the user is gathered, the profile is enriched to include additional weighted keywords associated with the concepts. These keywords are stored in subsidiary nodes, called *satellites*, linked to their associated concept nodes (planets). Links are also added between planets representing associations between concepts. Figure 2.3 shows an example excerpt of a user model based on this representation.

This representation was extended in WIFS [53], a filtering interface for personalizing results from the AltaVista [3] search engine. In this system, user profiles consist of three components: a header, including the user's personal data, a set of stereotypes, and a list of interests. A stereotype, or prototypical user, comprises a set of interests, represented by a frame of slots. Each slot contains three facets: *domain, topic*, and *weight*. The domain identifies an area of interest for the user, the topic is the specific term used by the user to identify the interest, and the weight indicates the user's degree of interest in the topic. The user model is represented as a frame containing the facets *semantic links* and *justification links*, as well as *domain*, *topic*, and *weight*. Figure 2.4 shows a sample profile based on this representation.

The semantic links include lists of keywords co-occurring in a document associated with the slot and having a degree of affinity with the topic. In this case, the profile is seen as a set of semantic networks, for which a slot is a planet and semantic links are the *satellites*. Figure 2.5 offers a simple example of just such a semantic network.

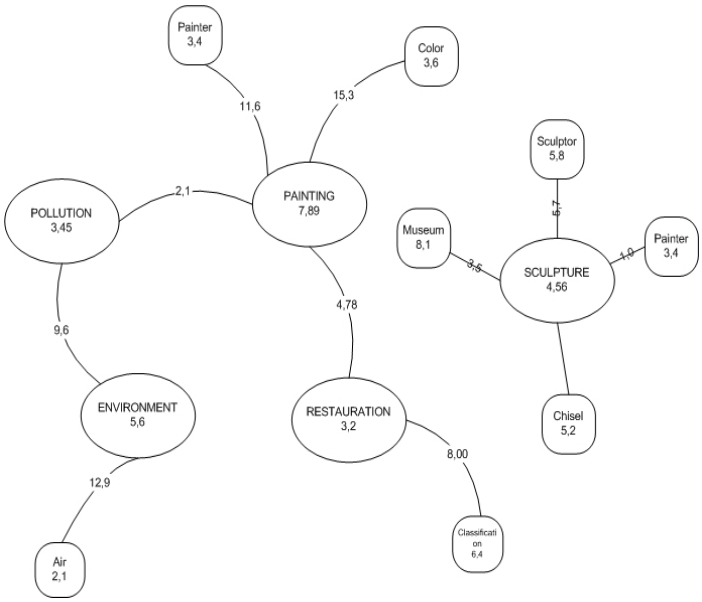

Fig. 2.3. An excerpt of a user profile based on semantic networks

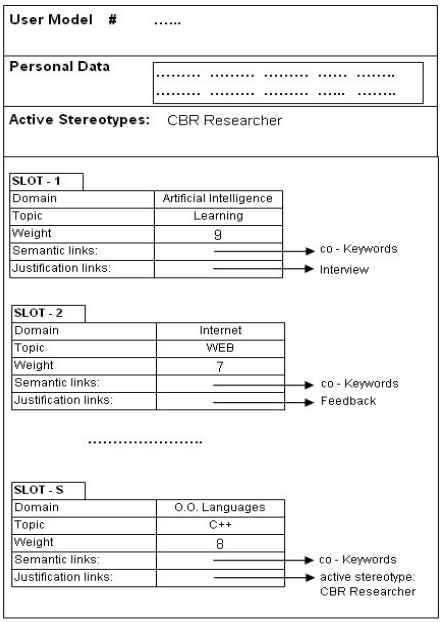

Fig. 2.4. An excerpt of user profile based on frames and semantic networks

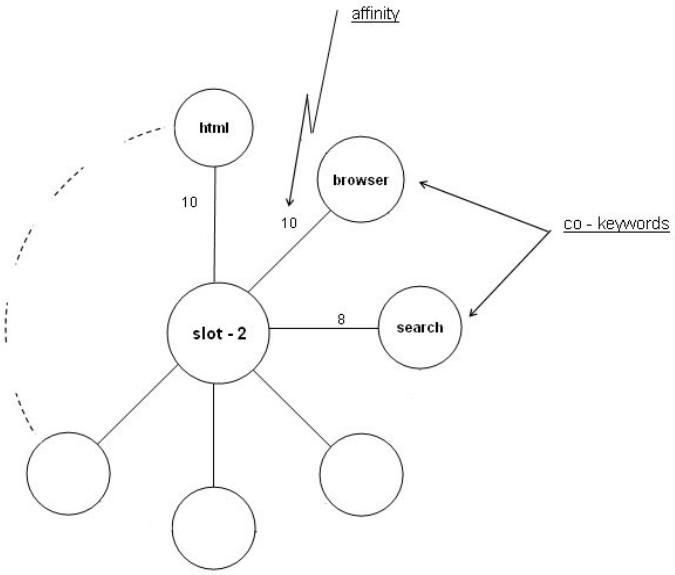

Fig. 2.5. An example of a semantic network

The justification links track down the reason why the slot to which they belong was inserted into the model. Their use is described in Section 2.4.2.

The system described in [25, 26] creates semantic network-based profiles that are used to model the interaction between users and information sources. Their Search of Associative Memory model tries to represent human memory, taking into consideration both the structure and the processes operating within it. The profile is organized into two main components, called the Long-Term Store (LTS) and the Short-Term Store (STS). Thus, each profile essentially consists of two keyword vectors, one that represents the long-term interests of the user (LTS) and another that represents the user's short-term interests. (STS). In particular, the STS identifies volatile information that is a subset of the LTS components. Links are created between words and context and each link is assigned a strength value that is used both in the learning and in the retrieval process. During the learning process, this value is calculated based on the amount of time each pair is temporarily stored in the STS.

2.3.3 Concept Profiles

Concept-based profiles are similar to semantic network-based profile in the sense that both are represented by conceptual nodes and relationships between those nodes. However, in concept-based profiles, the nodes represent abstract topics considered interesting to the user, rather than specific words or sets of related words. Concept profiles are also similar to keyword profiles in that often they are represented as

vectors of weighted features, but the features represent concepts rather than words or sets of words. Various mechanisms are applied to express how much the user is interested in each topic. The simplest technique is a numerical value, or weight, associated with each topic.

Bloedorn et al. [8] suggest using hierarchical concepts, rather than a flat set concepts, because this enables the system to make generalizations. The levels in the concept hierarchy can be fixed [99], or they can change dynamically according to the user's interests [15]. The simplest concept hierarchy based profiles are constructed from a reference taxonomy or thesaurus. More complex profiles may be constructed from reference ontologies. In the latter case, relationships between concepts are explicitly specified and the resulting profile may include richer information and a wide variety of relationship types.

Concept hierarchies were initially used to represent the content of Web pages [31, 42] but have more recently been used to represent user profiles. Most systems are based on a reference concept hierarchy, or taxonomy, from which a subset of the concepts and relationships are extracted and weighted to form a user profile. Because creating a broad and deep concept hierarchy is an expensive, mostly manual process, profiles are typically based on subsets of existing concept hierarchies. Conceptual search projects have used the *Sensus* ontology [31, 38], a taxonomy of approximately 70,000 nodes, and a subset of the *Yahoo!* directory [42, 111] as their reference conceptual hierarchies.

When using an existing directory as a source of concepts, certain transformations must take place to turn directory's contents into a concept hierarchy. Because the directory is designed to enable end-user browsing, not all parent-child links are conceptual. Some topics are split into children alphabetically, merely to partition the content. Others are split geographically. Some topics have dozens or hundreds of children whereas others may have few or none. Finally, some topics may have many Web pages linked to that subject whereas others may have little or no associated content. The profiling project must take these issues into consideration and decide which of the directory's subjects to include in the concept hierarchy. The more levels used, the more specific the user profile representation can become. However, if too many levels are used, general areas of interest may be lost. Often, non-conceptual parent-child subjects are removed and also those topics which have too few associated Web pages to act as examples for the profiling algorithm.

One of the first projects to build concept-based user profiles was the OBIWAN project [72]. Initially, they used a reference concept hierarchy containing 4,417 topics from the top four levels of the Magellan site. After the Magellan site ceased to exist, the group experimented with subject hierarchies downloaded from Yahoo! [111] and Lycos [48], eventually selecting the Open Directory Project (ODP) as a replacement, primarily because their directory is open source [65]. Initially, they represented profiles using 1,869 concepts from the top three levels of the ODP concept hierarchy [9] but, because the ODP has grown, they have used as many as 2,991 concepts from the top three levels [99]. Figure 2.6 shows an example of a conceptual user profile built from user's browsing histories by the OBIWAN project using the top three levels of the ODP [9,72,99].

Figure 2.7 shows the Web display of a particular user's profile in the misearch system [88]. This system builds the user profiles from implicit feedback collected via

search engine queries and clicked results. Users may view their top-weighted concepts with percentages that convey the relative weights of the concepts in the profile.

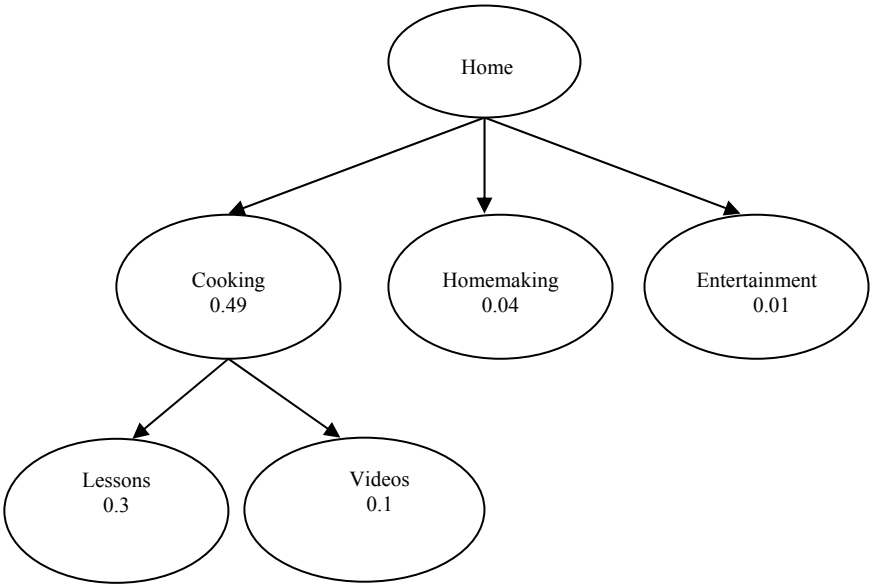

Fig. 2.6. An excerpt of a user profile based on concepts

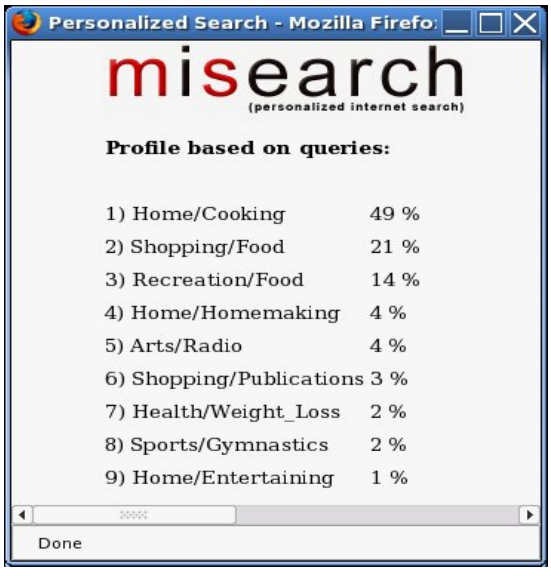

Fig. 2.7. A conceptual user profile representing a user interested in cooking, built from the top 3 levels of the ODP and the user's search history

The ODP is also used as the reference concept hierarchy in *Persona* [97]. However, because they use all concepts at any level in the ODP, they build more specific user profiles. These profiles contain only those concepts containing associated urls actually visited by the users, keeping the profile size scalable. On the other hand, because the Outride Personalized Search System [70] uses only 1,000 concepts from the Open Directory Project directory, their profiles are somewhat smaller than those used in OBIWAN and misearch, and they focus on capturing broad trends.

Although many of the previous projects may refer to their concept hierarchy as an ontology, the only relationship expressed is a parent-child relationship which generally represents an *is-a* and/or *has-a* relationship. The Semantic Web initiative is focusing on the creation and use of richer ontologies that can capture a wider variety of relationship types [7]. These ontologies are modeled using ontology representation languages such as *SHOE* [34, 47], *Extensible Markup Language* (XML) [106], the *Resource Description Framework* (RDF) [74], *RDF Schema* [75], *DAML+OIL* [19], or the *Web Ontology Language* (OWL) [101]. Some recent projects are exploring the use of these richer ontologies for improved search results [32, 112]. User profiles based on these richer ontologies may not be far away, however there remain serious roadblocks in the way, primarily due to scalability issues in creating large, diverse ontologies and exploiting them for searching large, distributed document collections. A comprehensive discussion of Semantic Web technologies for personalization can be found in Chapter 23 of this book [21].

2.4 User Profile Construction

User profiles are constructed from information sources using a variety of construction techniques based on machine learning or information retrieval. Depending on the user profile representation desired, different techniques may be appropriate. Techniques commonly used to construct keyword profiles are described in Section 2.4.1, whereas Section 2.4.2 and 2.4.3 describe construction techniques appropriate for semantic network profiles and concept profiles respectively. Profiles may be constructed manually by the users or experts, however, this is difficult and time consuming for most users and would be a barrier to widespread adoption of a personalized service. Techniques which automatically construct the profiles from user feedback are much more popular. Although some approaches use genetic algorithms or neural networks to learn the profiles, simpler, more efficient approaches based on probabilities or the vector space model are widely used and have been found to be effective in many applications.

No matter which construction method is chosen, the profile must be kept current to reflect the user's preferences accurately; this has proven to be a very challenging task [89]. Profile updating can be done automatically and/or manually. Automatic methods are preferred because it is less intrusive to the end user. Some authors warn against fully automatic profile updates, advising that user feedback, which requires minimal effort, should be used [90]. However, the results of experiments on fully automatic profile updating are promising [11, 12, 18, 72, 93].

2.4.1 Building Keyword Profiles

Keyword-based profiles are initially created by extracting keywords from Web pages collected from some information source, e.g., the user's browsing history or bookmarks. Some form of keyword weighting is done to identify the most important keywords from a given Web page, and often the number of words extracted from a single page is capped so that only the top *N* most highly weighted terms from any page contribute to the profile.

Table 2.2. Keyword Profile Construction Techniques

Profile Representation	Information Source	Construction Technique	Example
Single Keyword Vector	Web pages Implicit, positive feedback	Extract top-weighted keywords	Amalthaea [61]
One Keyword Vector per Interest	Web pages Explicit, positive feedback	Create document vector Compare interest vectors Merge closest interest vectors	WebMate [13]
Multiple Keyword Vectors per Interest	Web pages Explicit, positive and negative feedback	Create document vector Compare to interest vectors Add to closest match	Alipes [103]

The simplest type of profiling construction technique produces a single keyword profile for each user. *Amalthaea* [61] is one of many systems that creates profiles by extracting keywords from Web pages. They weight the keywords with the widely used tf*idf weighting scheme from information retrieval [80]. This project is somewhat unique in that it employs a learning algorithm based on genetic algorithms to adapt and expand the user profiles. In addition to the tf*idf weighting scheme, other projects have explored using Latent Semantic Indexing (LSI) [20] and Linear Least Squares Fit (LLSF) [45] for creating the keyword-based feature vectors.

Building multiple keyword profiles for each user, one per interest area, creates a more accurate picture of the user. Consider a user interested in Sports and Cooking. A single keyword vector will point towards the middle of these two topics, creating a picture of a user fascinated in athletes who cook, or people who cook for Superbowl parties. In contrast, by using a pair of vectors, the user profile more accurately represents the user's two independent interests.

WebMate [13] is an example of a system that builds user profiles that contain multiple keyword vectors, one per interest. Users provide explicit feedback on Web

pages they view as they browse. Document vectors are created by extracting keywords from the Web pages that receive positive feedback. Stop words, very common words such as 'and' and 'or', are removed and light stemming, removal of common word suffixes, is done to decrease the vocabulary size. Words are weighted using the tf*idf method common in vector space approaches. Title and heading words are specifically identified and weighted more highly. Unlike other systems that require the user to explicitly label interesting documents with their area of interest, *WebMate* automatically learns the interest areas. The learning algorithm is supplied with a fixed number of desired interests, *N*. The first *N* positive examples are each assumed to be a unique interest, and the vector for each document is used as an interest vector. Once there are more than *N* positive examples supplied, the two most similar interest vectors, as determined by the cosine similarity metric [80], are combined into a single interest vector. Figure 2.8 shows the creation of an user profile in *WebMate*.

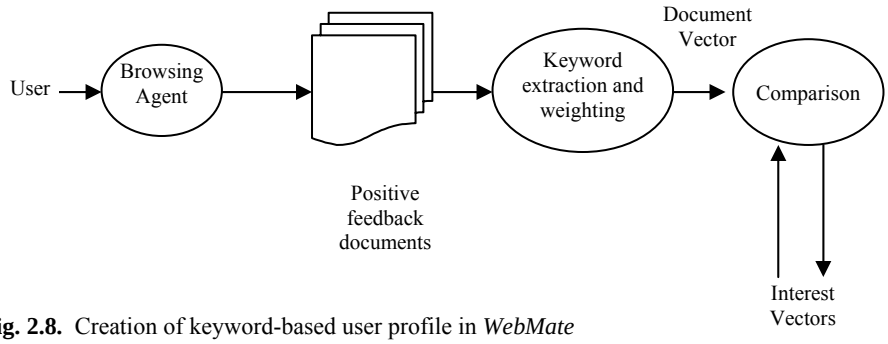

Fig. 2.8. Creation of keyword-based user profile in *WebMate*

Alipes [103] also creates user profiles that are based upon interest vectors, however they use multiple vectors per interest. In their case, each interest is modeled by three keyword vectors: long-term; short-term (positive), and short-term (negative). They consider negative feedback in addition to positive feedback, and the learning rate is affected by the strength of the user's preference. Like *WebMate*, they also automatically learn the user's interests, however, they base the creation of new interests on a similarity threshold rather than on a fixed number of desired interests. When a document vector is added to the user profile, it is compared to each of the three vectors for each interest using the cosine similarity metric. If the similarity exceeds a threshold, the document vector is added to the best matching interest. The strength of the user's feedback affects the amount of contribution the new document makes to the short-term vector, but the contribution to the long-term vector is determined by the number of example documents that have been learned so far, with the contribution factor declining over time. If, however, there is no match of sufficient strength between the document vector and the existing interest vectors, then a new interest is created and seeded with the document vector.

2.4.2 Building Semantic Network Profiles

Semantic network-based profiles are typically built by collecting explicit positive and/or negative feedback from users. Similar to keyword vector profile construction techniques, keywords are extracted from the user-rated pages. The techniques differ from those in the previous section because, rather than adding the extracted keywords to a vector, the keywords are added to a network of nodes. The nodes may represent individual words or, in more sophisticated approaches, a particular concept and its associated words. The terms "concepts" and "interests" are often used interchangeably in the literature. In this section, concept refers to a specific fine-grained idea and a collection of associated words, e.g., *dog* and its synonyms, whereas interest refers to higher level topics of interest to a user, e.g., *Animal Rights*, which in turn may be represented by a collection of associated concepts.

Semantic user profiles have an advantage over keyword-based profiles because they can explicitly model the relationship between particular words and higher-level concepts. Thus, they can deal more effectively with the inherent ambiguity and synonymy of natural language. However, this also places a barrier to the ease of constructing such system. They must either exploit an existing mapping between words and concepts, for example WordNet used by SiteIF, or they must build this through a learning mechanism as done by ifWeb [4], PIN [96], and InfoWeb [28], or they must build this manually, as is done in WIFS [53].

Table 2.3. Semantic Network Profile Construction Techniques

Profile Representation	Information Source	Construction Technique	Example
Single Semantic Network: One Node per Word	Sample documents Web pages Explicit positive and negative feedback	Extract top weighted words Create one node in semantic network per word Link nodes when the words they contain co-occur in documents	ifWeb [4]
Single Semantic Network: One Node per Concept	Web pages Implicit positive feedback	Extract top weighted words Map words into concepts using WordNet	SiteIF [92]
Single Semantic Network: One Node per Concept	Web pages Explicit positive feedback	Extract nouns Learn concepts using neural networks	PIN [96]
Single Semantic	Collection of	Create concept	InfoWeb [28]

Network: One Planet per Concept, One Satellite per Word	stereotype documents Explicit positive and negative feedback Direct user refinement	nodes from explicit feedback Add keyword nodes and arcs by refinement	
One Semantic Network per Interest: One Planet per Interest, One Satellite per Word	Collection of stereotype documents User Interview Explicit User Feedback, Direct Manipulation	Create concept nodes and keyword nodes using human experts Add concept nodes, keyword nodes, and arcs by refinement	WIFS [53]

In the simplest systems, each user is represented by a single semantic network in which each node contains a single keyword. The ifWeb system [4] initially builds this type of profile by presenting the user with a pre-determined small set of documents (4-6) and collecting positive and negative feedback on these documents. The profile is then refined as the user browses via a browsing agent and provides further feedback on Web pages proposed by ifWeb. Keywords are extracted from each of the pages receiving user feedback. These keywords undergo standard preprocessing, i.e., segmentation, stopword removal, stemming, and weighting. Moreover, keywords that occur too few times in a document, compared to a given threshold, are excluded. These keywords are then submitted to the IFTool subsytem [56] that is in charge of updating the semantic network representing the user profile. Keywords are added to the semantic network in which each node represents a keyword and the arcs represent the co-occurrence of the keywords within a document. If the keyword is already present in the semantic network, that node's score is increased or decreased, according to user feedback. If the keyword does not already appear, then a new node is created. Finally, the set of keywords extracted from the document are used to update the weights on the co-occurrence arcs. The IFTool Linguistic Processor is used both for user profile construction and for document evaluation. When the document is to be evaluated, IFTool extracts the information about its structure and its content which is used in order to build the document internal representation. The ifWeb system is able to consider different formats of documents such as HTML, PDF, plain text and postscript documents. The analysis of these documents is performed by a syntax-directed parser for each document format and modules for segmentation, stopword removal, stemming, and contextual weighting. By comparing a browsed document to the user model using user-settable criteria, the ifWeb system classifies the document in one of the three categories ‘interesting,’ ‘not-interesting,’ or ‘indifferent.’

Because the same concept can be expressed using many different words, semantic network profiles in which the nodes represent concepts, rather than individual words, are likely to be more accurate. The SiteIF [92] system builds this type of semantic

network-based profile from implicit user feedback. Essentially, the nodes are created by extracting concepts from a large, pre-existing collection of concepts, WordNet [105]. As the user browses the Web, representative keywords are extracted from documents using the same process as the ifWeb system. These keywords are mapped into concepts using WordNet, a collection of 100,000 word forms organized into 80,000 *synsets*. Polysemous words are then disambiguated by analyzing their synsets to identify the most likely sense given the other words in the document. Finally, the synsets for the disambiguated representative keywords are combined to yield a user profile that is a semantic net whose nodes are concepts (synsets) and whose arcs represent the co-occurrence of two concepts within a document. Every node and every arc has a weight that represents the user's level of interest. In order to capture a shift in the user's interests over time, the weights in the network are periodically reconsidered and possibly lowered, depending on the time passed from the last update. Also, nodes and arcs that are no longer useful may be removed from the net.

When building user profiles based on semantic networks of concepts, the personalized system may not be able to afford the time and space overhead necessary to incorporate a large concept network. In other situations, there may not exist appropriate online collections of concepts from which nodes can be created. This may happen when the user's interests are domain specific, or based on recently created topics. PIN [96] is a personalized news system that also builds semantic network based user profiles in which the nodes represent concepts. Each node consists of an interest term, essentially the name of the concept, and a set of keywords related to that concept. Explicit user feedback is collected via user-supplied ratings of news articles. Because news stories, by definition, often cover previously unknown topics, PIN learns its concepts from the examples rather than mapping its examples to existing concepts. The profile is initially instantiated by the user supplying one or more concepts, consisting of an interest term and one or more keywords. For each positively rated article, a morphological analyzer is used to identify the part-of-speech of each word. To reduce complexity by reducing the number of features supplied to the neural network, only nouns are extracted. These are reduced to their stem, and the profile is searched to see if that keyword already appears in the list of keywords for the existing interest. If the keyword is found, the count for that keyword is incremented. The user feedback also sends the positively rated article to ARAM [95], a neural network based learning system. Through a refinement process, ARAM [95] learns new concepts that are not explicitly mentioned by the user, enhancing the user profile. The learned concepts are weighted according to the rating given to the Web page by the user.

InfoWeb [28], a query expansion and document filtering system for an online digital library, also builds user profiles represented as a semantic network of concepts. The profiles are built based on explicit user feedback collected by a browsing agent. One problem with explicit feedback techniques is the time and effort required from the users. By carefully selecting the documents on which the user feedback is collected, rather than presenting the user with random documents, they are able to collect representative information with less user effort. In order to identify documents that represent the scope of the digital library, they cluster the documents in the collection into a pre-determined set of k possible categories. To ensure that the categories created by the clustering algorithm are semantically meaningful, a domain

expert chooses *k* documents that are representative of the *k* semantic categories into which the expert divides the collection. These documents are then used as the seeds for the clustering algorithm. After the clustering, the document closest to the centroid of the cluster, is selected to act as the representative document for the cluster, called the *stereotype.*

To build their initial profile, users are asked to give explicit relevance feedback (both positive and negative) about the stereotypes. Once the feedback is collected, the rated documents are processed to extract the highest weighted keywords, creating a single semantic network representation of the user profile. Initially, each extracted keyword from the rated stereotype is represented as a planet. Implicit feedback from users, or direct user manipulation of the profile, is used to maintain and refine the semantic network. As the user interacts with the system, feedback documents are matched to the profile based on a linear combination of the individual terms in the document and the user's semantic network, similar to that used in Rocchio classification [79]. This feedback is used to enrich the profile – when the weights of the links exceed a certain threshold – by adding satellite nodes linked to the appropriate planet node. User feedback is also used to create links between concepts, viewed as creating links between planets. These links are added and updated according to a distance metric calculated between the terms in the documents.

In the aforesaid approaches, a single semantic network is built to represent the user's interests. This approach suffers from the same lack of accuracy of systems that build a single keyword vector per user. It is generally preferable to produce a more fine-grained representation of user profiles as collections of interests, each represented by a semantic network. This approach is used by WIFS [53], an extension to InfoWeb [28] applied to searching the World Wide Web. Since this system is applied to Web search, it is not as easy to create the initial set of topics from which user interests can be selected. The Web is far too large to cluster its contents so as to determine all possible topics. Instead, the WIFS system features a preliminary work led by human experts, who identified the set of terms, stored in a Terms Data Base, deemed most relevant for each specific field of interest. Besides, these experts set a basic level of knowledge of stereotypes, each one representing the prototype user's information needs. The first time a user starts a working session, he is interviewed by the system to obtain a first set of information needs. This information is used to determine the stereotype(s) (named *active stereotypes*) that best approximate his information needs. The user model is initialized with the information provided in the interview and with the data inherited from the active stereotypes.

As the user interacts with WIFS [53], his relevance feedback is used to refine the profile by adding, updating, or removing planets, satellites and affinities. There are five different ways to update the user model, four automatic and one manual. Firstly, the user's current interest, chosen from their existing interests, is updated according to the match between keywords in the semantic networks for each interest and keywords extracted from the currently viewed Web page and the query used to locate that page. Secondly, occurrences of keywords in the Web page that are already contained in the semantic network for the current interest are used to modify the affinity value of the arc between the satellite node, for the keyword, and the planet node, for the interest. The increase is proportional to the feedback value supplied by the user. If the new value does not fall within a predefined range, then the keyword node is removed from

the network. Thirdly, new keywords extracted from the Web page are used to add new satellite nodes to the semantic network for the current interest. The value of the user's feedback on the page is used to set the weights on the arcs (affinity value) linking the new satellite nodes to the planet node whose topic pertains to the document; in the way the system maintains the term co-occurrence information. Fourthly, if the feedback value on the Web page exceeds a threshold, keywords extracted from the Web page can be used to add a new topic with the term's name and the domain = "filler". Finally, users are also able to explicitly modify their personal profile through direct manipulation. The consistency of the model is maintained by a simple justification-based truth maintenance system (JTMS) that uses the justification links described in Figure 2.4. They help provide explanations of the reason why the slot was inserted into the model and the evaluation of its weight. Such links maintain the consistency of the model, should the cause be removed. For example, if a shift in the user's interests occurs during the course of an interaction, thereby changing the active stereotype, the slots previously justified by the older stereotype will be eliminated.

2.4.3 Building Concept Profiles

This section describes three representative systems that build user profiles represented as weighted concept hierarchies. Although each uses a different construction methodology, they each use a reference taxonomy as the basis of the profile. These profiles differ from semantic network profiles because they describe the profiles in terms of pre-existing concepts, rather than modeling the concepts as part of the user profile itself. Thus, they all require some way of determining which concepts a user is interested in based on their feedback. Although some systems collect feedback on pre-classified documents, many collect feedback on a wide variety of documents then do text classification to identify the concepts exemplified by each document. Many research projects in this section refer to their concept hierarchies as *ontologies*. However, in this section, we use the term concept hierarchy when the ontology contains only "is-a" links, and restrict the use of the word ontology to (future) systems that support a rich variety of relationships between the concepts, including logical propositions that formally describe the relationship.

Table 2.4. Concept Profile Construction Techniques

Reference Taxonomy	Information Source	Construction Technique	Example
Open Directory Project All Concepts	Explicit positive feedback on pre-classified Web pages	Tree-coloring	Persona [97]
Yahoo!	Implicit positive feedback on Web pages and	Clustering	ARCH [86]

	search results		
CORA 97 Concepts	Implicit and explicit positive feedback on pre-classified research papers	Tree-coloring Propagation to parent concepts	Foxtrot [55]
Open Directory Project ~2,000 Concepts	Implicit positive feedback on any Web page[1] or queries and search results[2]	Text classification to identify concepts	OBIWAN [72] Misearch [87]
Open Directory Project 619 Concepts	Implicit positive feedback using queries, search results; explicit positive feedback on categories	Text classification to identify concepts Expand classifier training based on feedback	Liu et al [45]
Open Directory Project 55 Concepts	Implicit positive feedback on any Web page	Text classification to identify concepts Taxonomy adapts to add/remove concepts	PVA [15]
ACM Topic Hierarchy 1,287 Concepts	Implicit feedback via bibliography contents, queries Explicit feedback via profile manipulation	Tree coloring Direct manipulation Recommendations	Bibster [33]

Persona [97] is exploring personalized search that exploits user profiles represented as a collection of weighted concepts based upon the Open Directory Project's concept hierarchy. The system builds a taxonomy of user interest and disinterest using a tree coloring method. As the user searches the collection of pre-classified documents in the ODP, they are asked to provide explicit feedback on the resulting pages. This feedback is then used to update their profile. Since the pages are already manually mapped into the ODP concepts, the user profile can be easily updated by keeping a count of the number of times a given concept was visited, i.e., had a page viewed by the user, and the number of positive and negative feedbacks the node received, and the set of urls associated with the node. Because the system uses

pre-classified documents, the profile is able to contain any or all concepts in the ODP and the mapping of visited pages to concepts is very accurate. One difficulty with this approach is that, because the ODP hierarchy is so deep and contains so many concepts, the profile can become very large and contain many very narrow concepts. When using this profile to provide personalized services, matching may need to be done using the parent, grandparent, or higher level ancestors of colored concepts, and deciding the level at which to perform matching remains to be investigated.

The ARCH system [86] is a hybrid approach combining keyword vector based user profiles with a concept hierarchy. The system collects implicit user feedback in the form of browsing and search activity in order to identify a set of documents in which the user has shown interest. These documents are clustered to identify their areas of interest and the centroid for each cluster is calculated, producing a weighted keyword vector representing that interest. The authors expand upon the keyword vector approaches described in Section 2.4.1, however, by mapping between user interests and concepts in the Yahoo! concept hierarchy. For each Yahoo! concept, the system calculates the centroid of a set of training documents. When a the user enters a query, they identify the most similar interest vector, then calculate the similarity between that interest vector and the concept vectors to find the most similar concept. Terms from the top-matching concept are then used for query expansion. Although this is an interesting approach, the other projects that explicitly model user profiles as collections of weighted concepts are computationally more efficient and likely to be as accurate as this hybrid.

Persona builds its profiles from manually classified documents and ARCH employs clustering to identify user interests. All of the other systems in this section rely on text classification in order to map the information collected about the user into the appropriate concept(s) in a concept hierarchy. This approach seems quite robust, but hinges on the quality of the information used to train the text classifier, the match between the feedback documents and the training documents, and the accuracy of the classifier. Text classification is a supervised approach that attempts to assign documents to the best matching concept(s) from a predefined set of concepts. It is comprised of two phases: learning and classification. In the learning phase, the system is given a series of documents classified by hand, and it attempts to acquire enough information from them in order to classify a new document. In the classification phase, the system receives a new document and assigns it a concept label based on its match with the training data. Several methods for text classification have been developed, each with a different approach for comparing the new documents to the reference set. These include comparisons between a variety of frequently-used vector representations of the documents (Support Vector Machines, k-nearest neighbor, linear least-squares fit, tf * idf); the use of the joint probabilities of the words being in the same document (Naive Bayesian); decision trees; and neural networks. A very complete survey and comparison of such methods is presented in [108], and more are discussed in [67, 77, 80]. Recent approaches focus on extensions of traditional classification approaches to hierarchical concept hierarchies [23, 57].

The Foxtrot recommender system for a digital library of computer science papers [55] takes a similar approach to Persona. The authors organize the library contents into a concept hierarchy of 97 classes [51], and manually provide 5-10 example documents for each concept. They employ text classification techniques to

automatically classify the remainder of the papers in the library. As users interact with an online digital library of pre-classified research papers, implicit (browsing activity) and explicit feedback (relevance judgments) is collected. The concepts associated with the documents are used to update the concept profile, with explicit feedback contributing 10 times more to a concept's weight than implicit feedback. The system also includes a linear time-delay factor so that, as days go by, previously contributed papers contribute less and less to the weight of the concepts in the profile. This may not be a necessary enhancement to the algorithm since, as more feedback is collected, concepts that are no longer of interest will cease to grow whereas concepts for current interests will continue to grow and the relative weights of the past and current interests will shift. The time-delay factor merely accelerates this process. The system also propagates the 50% of the weight of low-level concepts to their parent concepts. This is an important enhancement, allowing the profile to represent the fact that a user interested in, for example, "Machine Learning" and "Data Mining" is also interested in the parent concept, "Artificial Intelligence."

The OBIWAN project represents user profiles as a weighted concept hierarchy built from a reference concept hierarchy. The profile creation process in OBIWAN is shown in Figure 2.9. Initially, the Magellan directory was used [71], but more recently the Open Directory Project has been adopted [9]. The main difference between this approach and Persona is that the system is not restricted to building the user profiles from pre-classified documents. Any source of representative text may be automatically classified by the system to find the best matching concepts from the ODP, and then those concepts have their weights increased. The project focuses on personalized search and navigation as a way to validate the quality of the profiles produced. The authors have built the profiles based upon browsing histories submitted as browser caches [27], collected by proxy servers [98], or captured from desktop screens [10]. Most recently, this approach has been used in the misearch project [87] that builds the profiles by collecting and classifying the user search histories rather than the user's browsing history. Unlike Persona, which requires explicit feedback from the user, the profiles are built from implicit information, the queries submitted and snippets of search results clicked on by the user. These profiles, regardless of which information source has been used, have been shown to be able to statistically significantly improve personalized search.

Similar to the OBIWAN project, Liu et al [45] construct user profiles based on ODP categories using text classification. The authors use only the top 2 levels in their user profile, creating a broader overview of user interests. The system trains the classifier for the ODP categories using words extracted from the ODP associated documents, but, because they enhance the classifier with terms extracted from user-supplied feedback, their profiles will be less sensitive to the terminology contained in the ODP training documents. Because the system requires users to explicitly indicate the categories of interest, this approach is not entirely based on implicit feedback.

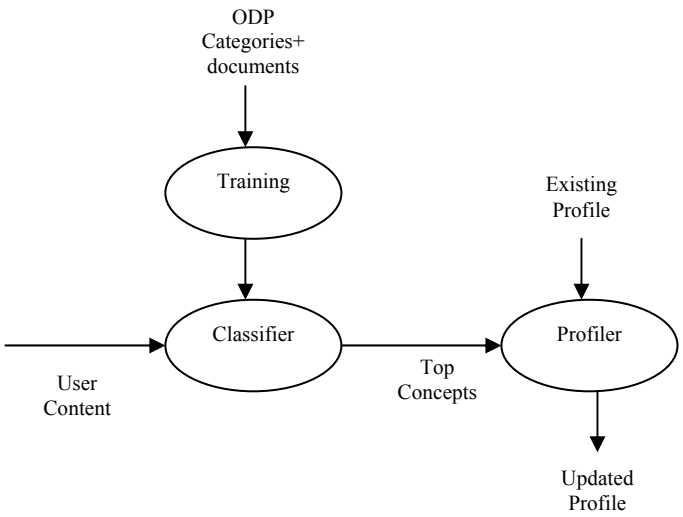

Fig. 2.9. User profile creation in OBIWAN

PVA [15] also builds a user profile represented as a weighted concept hierarchy from implicit feedback, in their case, information collected from a proxy server. Similar to OBIWAN, they automatically classify the representative documents for a user into a pre-existing concept hierarchy. Currently, the system uses a three-level concept hierarchy containing 55 concepts used by the Yam [107] search site. Because the authors are focusing a personalized recommendation of news articles, a fast changing domain, the system retrains the classifier daily with manually-classified new content collected from three news agencies. PVA differs from previous projects, however, in that it does not apply a variation of a tree coloring algorithm. Rather, the system uses the concepts returned by the classifier to dynamically grow and shrink the user profile. Initially, the user profile is represented just as the collection of top level concepts in the hierarchy. The results of the classification are therefore used to increase the weight, which they call Energy, of the appropriate top-level concept, and the full classification path for each classified document. For example, if a document is classified into "/Sports/Basketball/NBA", the Energy for "/Sports" will be increased. As documents are classified over a period of time, the Energy value for a concept may increase beyond a threshold. When this occurs, the system then splits that concept into two concepts based upon the classification paths of the contained documents. Thus, the profile grows as more information is collected about a given user. The authors also incorporate an aging process in their user profiling method. As time passes, the contribution a document makes to its associated concept's Energy value decreases. If the total Energy for a concept falls below a threshold, that concept is merged back into its parent concept. Thus, user profiles can grow and shrink to better reflect the user's interests.

As user profiles begin to move out from the research world and into use, the need to keep them accurate over time increases. PVA represents an entirely automatic approach to profile adaptation. In contrast, Bibster [33] employs collaborative

feedback from multiple users with similar interests in order to recommend profile changes to users with similar users. Although the authors discuss users as having ontologies, essentially each user is represented by a concept profile, a weighted set of concepts selected from the ACM Topic Hierarchy. Users manually manipulate their profiles by adding/removing concepts and implicit feedback from the user's personal bibliography of documents (which are pre-categorized with respect to the concept hierarchy) and as they search. The unique feature of this system is that, as one user manipulates/updates their profile, these changes are also suggested to other users with similar profiles. The advantage of a collaborative approach is that, when many similar people are providing feedback, less feedback per individual is needed in order to construct and maintain an accurate profile. The drawback is that, when many people's feedback is combined, the fit between the profile and a particular individual may not be as good. The authors circumvent this process by presenting prospective changes to the profile as recommendations that are subject to user review before being adopted in their personal profile.

2.5 Conclusions

In conclusion, there is a tremendous growth in the approaches taken to represent, construct, and employ user profiles. These enabling technologies are key to providing users with accurate, personalized information services. There are a variety of techniques being investigated, but implicitly-created profiles place less burden on the user and, in several instances, seem to be able to adequately capture the user's interests. As these technologies mature, we see a move from simple keyword vectors to richer, conceptual representations. In future, profiles will also need to incorporate temporal and contextual information such as: What is the user doing now? What information has the user already seen? Where is the user located? However, personalized services are becoming a reality as user profiles move from the laboratory to the Internet.

References

1. The Acxiom Corporation, http://www.acxiom.com/ (last access on February 2006)
2. Adar, E., Karger, D.: Haystack: Per-User Information Environments. In: Proceedings of the 8th International Conference on Information Knowledge Management (CIKM), Kansas City, Missouri, November 2-6 (1999) 413-422
3. Altavista search engine, http://www.altavista.com/ (last access on October 2005)
4. Asnicar, F., Tasso, C.: ifWeb: A Prototype of User Model-Based Intelligent Agent for Documentation Filtering and Navigation in the World Wide Web. In: Proceedings of the 6th International Conference on User Modeling, Chia Laguna, Sardinia, Italy, June 2-5 (1997) 3-11
5. Balabanovic, M., Shoham, Y.: Fab: Content-Based Collaborative Recommendation. In: Communications of the ACM, 40(3), March (1997), 66-72

6. Barrett, R., Maglio, P., Kellem, D.C.: How to Personalize the Web. In: Proceedings of the SIGCHI conference on Human factors in computing systems, Atlanta, March 22-27 (1997) 75-82
7. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 284(5) May (2001) 34–43
8. Bloedorn, E., Mani, I., MacMillan, T.R.: Machine Learning of User Profiles: Representational Issues. In: Proceedings of AAAI 96, IAAA 96, Portland, Oregon, August 4-8, (1) (1996) 433-438
9. Chaffee, J., Gauch, S.: Personal Ontologies for Web Navigation. In: Proceedings of the 9th International Conference On Information Knowledge Management (CIKM), Washington, DC, November 6-11 (2000) 227-234
10. Challam, V., Gauch, S.: Contextual Information Retrieval Using Ontology Based User Profiles. In: ACM Transactions on Internet Technologies (pending)
11. Chan, K.P.: A Non-Invasive Learning Approach to Building User Web Profiles. In: Proceedings of the KDD-99 Workshop on Web Usage Analysis and User Profiling, San Diego, August 15-18 (1999) 39-55 (last access on October 2006) http://citeseer.ist.psu.edu/chan99noninvasive.html
12. Chan, K.P.: Constructing Web User Profiles: A Non-Invasive Learning Approach. In: Web Usage Analysis and User Profiling, LNAI 1836, Springer- Verlag (2000) 39-55
13. Chen, L., Sycara, K.: A Personal Agent for Browsing and Searching. In: Proceedings of the 2nd International Conference on Autonomous Agents, Minneapolis/St. Paul, May 9-13, (1998) 132-139
14. Chen, Y-S., Shahabi, C.: Automatically improving the accuracy of user profiles with genetic algorithm. In: Proceedings of IASTED International Conference on Artificial Intelligence and Soft Computing, Cancun, Mexico, May 21-24 (2001) 283-288
15. Chen, C., Chen, M., Sun, Y.: PVA: A self-adaptive Personal View Agent. Journal of Intelligent Information Systems, 18 (2-3), March-May (2002) 173-194
16. Chesnais, P., Mucklo, M., Sheena, J.: The Fishwrap Personalized News System. In: Proceedings of IEEE 2nd International Workshop on Community Networking: Integrating Multimedia Services to the Home, Princeton, NJ, June 20-22 (1995) 275-282
17. Chien, W.: Learning Query Behavior In the Haystack System. Master's thesis, MIT, June (2000)
18. Crabtree, B., Soltysiak, S.: Identifying and Tracking Changing Interests. International Journal on Digital Libraries, 2(1), (1998), 38-53
19. The DARPA Agent Markup Language Homepage, http://www.daml.org/ (last access on October 2005)
20. Deerwester, S., Dumais, S., Furnas, G., Landauer, T.K., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science, 41(6) (1990) 391-407
21. Dolog, P., and Nejdl, W.: Semantic Web Technologies for the Adaptive Web. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
22. Dumais, S., Cutrell, E., Cadiz, J.J., Jancke, G., Sarin, R., Robbins, D.C.: Stuff I've seen: a system for personal information retrieval and re-use. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, Toronto, Canada, July 28 - August 01, (2003) 72-79
23. Dumais, S., Chen, H.: Hierarchical classification of Web content. In: Proceedings of the 23rd ACM International Conference on Research and Development in Information Retrieval, (2000) 256-263
24. Epinions website, http://www.epinions.com/ (last access on February, 2006)
25. Gasparetti, F.: Adaptive Web Search: User Modeling based on Associative Memory and

Multi-Agent Focused Crawling. PhD Thesis, University of Roma Tre, 2005
26. Gasparetti, F., Micarelli, A.: User Profile Generation Based on a Memory Retrieval Theory. In: The 1st International Workshop on Web Personalization, Recommender Systems and Intelligent User Interfaces (WPRSIUI 2005), Reading, UK, October 3-7 (2005) http://citeseer.ist.psu.edu/gasparetti05user.html
27. Gauch, S., Chaffee, J., Pretschner, A.: Ontology-Based User Profiles for Search and Browsing. In: Web Intelligence and Agent Systems 1(3-4) (2003) 219-234
28. Gentili, G., Micarelli, A., Sciarrone, F.: Infoweb: An Adaptive Information Filtering System for the Cultural Heritage Domain. Applied Artificial Intelligence 17(8-9) (2003) 715-744
29. Google Desktop, http://desktop.google.com/ (last access on October 2005)
30. Google Personalized Search, https://www.google.com/psearch/ (last access on September 2005)
31. Guarino, N., Masolo, C., Vetere, G.: OntoSeek: Content-Based Access to the Web. IEEE Intelligent Systems, May 14(3) (1999) 70-80
32. Guha, R., McCool, R., Miller, E.: Semantic Search. In: Proceedings of the WWW2003, Budapest, Hungary, May 20-24 (2003) 700-709
33. Haase, P., Hotho, A., Schmidt-Thieme, L., Sure, Y.: Collaborative and Usage-driven evolution of personalized ontologies. In: Proceedings of the 2nd European Semantic Web Conference, Heraklion, Greece, May 29-June 1 (2005) 486-499
34. Heflin, J., Hendler, J., Luke S.: SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), University of Maryland at College Park (1999) (last access on September 2005) http://www.cs.umd.edu/projects/plus/SHOE/pubs/techrpt99.pdf
35. Hoashi, K., Matsumoto, K., Inoue, N., Hashimoto, K.: Document Filtering method using non-relevant information profile. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Athens, Greece, July 24-28 (2000) 176-183
36. Kelly, D., Teevan, J.: Implicit feedback for inferring user preference: a bibliography. ACM SIGIR Forum 37(2) (2003) 18-28
37. Kim, H., Chan, P.: Learning implicit user interest hierarchy for context in personalization. In: Proceedings of IUI' 03, Miami, Florida, January 12-15 (2003) 101-108
38. Knight, K., Luk, S.: Building a Large Knowledge Base for Machine Translation. In: Proceedings of American Association of Artificial Intelligence Conference (AAAI), Orlando, Florida, July 18–22 (1999) 773- 778
39. Kobsa, A.: Privacy-Enhanced Web Personalization. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
40. Kobsa, A.: Generic User Modeling Systems. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
41. Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying Collaborative Filtering To Usenet News. Communications of the ACM 40(3) (1997) 77-87
42. Labrou, Y., Finin, T. : Yahoo! As An Ontology – Using Yahoo! Categories To Describe Documents. In: Proceedings of the 8th International Conference On Information Knowledge Management (CIKM), Kansas City, Missouri, November 2-6 (1999) 180-187
43. Lieberman, H.: Letizia: An Agent That Assists Web Browsing. In: Proceedings of the 14th International Joint Conference On Artificial Intelligence, Montreal, Canada, August (1995) 924-929
44. Lieberman, H.: Autonomous Interface Agents. In: Proceedings of the ACM Conference on Computers and Human Interaction (CHI'97) Atlanta, Georgia, March 22-27 (1997) 67 - 74

45. Liu, F., Yu, C., Meng, W.: Personalized web search by mapping user queries to categories. In: Proceedings CIKM'02, Mclean, Virginia, November 4-9 (2002) 558-565
46. Liu, F., Yu, C., Meng, W.: Personalized Web Search For Improving Retrieval Effectiveness. In: IEEE Transactions on Knowledge and Data Engineering, 16(1), January (2004) 28-40
47. Luke, S., Spector, L., Rager, D., Hendler, J.: Ontology-Based Web Agents. In: Proceedings of the First International Conference on Autonomous Agents (AA'97) Association for Computing Machinery, California, February 5-8 (1997) 59-66
48. Lycos, http://www.lycos.com (last access on September 2005)
49. Malone, T., Grant, K., Turbak, F., Brobst, S., Cohen, M.: Intelligent Information Sharing Systems. Communications of the ACM 30(5) (1987) 390-402
50. Marais, H., Bharat, K.: Supporting cooperative and personal surfing with a desktop assistant. In: Proceedings of ACM UIST'97, Banff, Alberta, Canada, October 14-17 (1997) 129-138
51. McCallum, A.K., Nigam, K., Rennie, J., Seymore, K.: Automating the Construction of Internet Portals with Machine Learning. In: Information Retrieval 3(2) (2000) 127-163
52. Micarelli, A., Gasparetti, F., Sciarrone, F., and Gauch S.: Personalized Search on the World Wide Web. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
53. Micarelli, A., Sciarrone, F.: Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. User Modeling and User-Adapted Interaction 14(2-3) June (2004) 159-200
54. Micarelli, A., Sciarronne, F., Marinilli, M.: Web Document Modeling. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
55. Middleton, S.E., Shadbolt, N.R., De Roure, D.C.: Capturing interest through inference and visualization: Ontological user profiling in recommender systems. In: International Conference on Knowledge Capture, K-CAP 2003, Sanibel Island, Florida, September (2003) 62-69 http://portal.acm.org/citation.cfm?id=945657
56. Minio, M., Tasso, C.: User Modeling for Information Filtering on INTERNET Services: Exploiting an Extended Version of the UMT Shell. In: UM96 Workshop on User Modeling for Information Filtering on the WWW; Kailua-Kona, Hawaii, January 2-5 (1996) http://ten.dimi.uniud.it/~tasso/UM-96UMT.html
57. Mladenic, D.: Turning Yahoo into an Automatic Web-Page Classifier. In: Proceedings of the 13th European Conference on Aritficial Intelligence ECAI (1998) 473-474
58. Mladenić, D.: Personal WebWatcher: Design and Implementation. Technical Report IJS-DP-7472, J. Stefan Institute, Department for Intelligent Systems, Ljubljana, Slovenia (1998) (last access on October 2006) http://www-ai.ijs.si/DunjaMladenic/papers/PWW/pwwTR.ps.Z
59. Mobasher, B.: Data Mining for Web Personalization. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
60. Montebello, M., Gray, W., Hurley, S.: A Personal Evolvable Advisor for WWW Knowledge-Based Systems. In: Proceedings of the 1998 International Database Engineering and Application Symposium (IDEAS'98), Cardiff, Wales, U.K, July 8-10 (1998) 224-233
61. Moukas, A.: Amalthaea: Information Discovery And Filtering Using A Multiagent Evolving Ecosystem. In: Applied Artificial Intelligence 11(5) (1997) 437-457
62. Netflix Website, http://www.netflix.com/ (last access on February, 2006)

63. Nichols, D.: Implicit Rating and Filtering. In: Proceedings of the 5th DELOS Workshop on Filtering and Collaborative Filtering, Budapest, November 10-12 (1998) 31-36 (last access on October 2006) http://citeseer.ist.psu.edu/nichols98implicit.html
64. Oard, D., Marchionini, G.: A Conceptual Framework for Text Filtering. Technical Report EE-TR-96-25 CAR-TR-830 CLIS-TR-9602 CS-TR-3643. University of Maryland, May (1996)
65. The Open Directory Project (ODP), http://dmoz.org (last access on September 2005)
66. Papazoglou, M.: Agent-oriented technology in support of e-business. In: Communications of the ACM, 44(4), April (2001) 71-77
67. Pazzani, M., Billsus, D.: Content-based recommendation systems.In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
68. Pazzani, M., Muramatsu, J., Billsus, D.: Syskill & Webert: Identifying Interesting Web Sites. In: Proceedings of the 13th National Conference On Artificial Intelligence Portland, Oregon, August 4–8 (1996) 54-61
69. Perkowitz, M., Etzioni, O.: Adaptive Web Sites: Automatically Synthesizing Web Pages. AAAI, Madison, Wisconsin, July 26–30 (1998) 727-732
70. Pitkow, J., Schütze, H., Cass T. et all.: Personalized search. CACM 45(9) (2002) 50-55
71. Pretschner, A.: Ontology Based Personalized Search. Master's thesis. University of Kansas, June (1999)
72. Pretschner, A., Gauch, S.: Ontology Based Personalized Search. In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence (ICTAI) November 8-10 (1999) 391-398
73. Quiroga, L., Mostafa, J.: Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems. In: D. H. Kraft (Ed.), Proceedings of the 63rd annual meeting of the American Society for Information Science and Technology, Medford, NJ: Information Today 37 (2000) 4-13
74. Resource Description Framework, http://www.w3.org/RDF/ (last access on October 2005)
75. Resource Description Framework Schema, http://www.w3.org/TR/rdf-schema/ (last access on October 2005)
76. Rich, E.: Users are Individuals: Individualizing User Models. In: International Journal of Man-Machine Studies 18 (1983) 199–214
77. Ruiz, M., Srinivasan, P.: Hierarchical Neural Networks For Text Categorization. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, California, August 15-19 (1999) 281-282
78. Sakagami, H., Kamba, T.: Learning Personal Preferences on Online Newspaper Articles From User Behaviors. In: Proceedings of the 6th International WWW Conference, Santa Clara, California, April 7-11 (1997) 291-300
79. Salton, G. Developments in automatic text retrieval. Science. Vol.253. Pages 974-979, 1991
80. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill (1983)
81. Schafer, J.B., Frankowski, D., Herlocker, J., and Sen, S.: Collaborative Filtering Recommender Systems. In Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization, Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) this volume
82. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1) (2002) 1–47
83. Seruku Toolbar, http://www.seruku.com/index.html (last access on October 2005)
84. Shavlik, J., Eliassi-Rad, T.: Intelligent Agents for Web-Based Tasks: An Advice-Taking Approach. In: Working Notes of the AAAI/ICML-98 Workshop on Learning for text

categorization. Madison, WI, July 26-27 (1998) (last access on October 2006) http://citeseer.ist.psu.edu/shavlik98intelligent.html
85. Shavlik, J., Calcari, S., Eliassi-Rad, T., Solock, J.: An Instructable, Adaptive Interface for Discovering and Monitoring Information on the World Wide Web. In: Proceedings of the 1999 International Conference on Intelligent User Interfaces. Redondo Beach, California, January 5-8 (1999) 157-160
86. Sheth, B.: A Learning Approach to Personalized Information Filtering. Master's thesis. Massachusetts Institute of Technology (1994)
87. Sieg, A., Mobasher, B., Burke, R.: Inferring users information context: Integrating user profiles and concept hierarchies. In: 2004 Meeting of the International Federation of Classification Societies, IFCS, Chicago, July (2004) http://maya.cs.depaul.edu/~mobasher/papers/arch-ifcs2004.pdf
88. Speretta, M., Gauch, S.: Personalized Search based on User Search Histories. In: IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). Compiegne University of Technology, France September 19-22 (2005) 622-628
89. Stadnyk, I., Kass, R.: Modeling User's Interests in Information Filters. Communications of the ACM, 35(12) December (1992), 49-50
90. Soltysiak, S.J., Crabtree, I.B.: Automatic Learning Of User Profiles - Towards the Personalization of Agent Services. BT Technology Journal 16 (3), July (1998) 110-117
91. Sorensen, H., McElligott, M.: PSUN: A Profiling System for Usenet News. In: Proceedings of CIKM'95 Workshop on Intelligent Information Agents, Baltimore Maryland, December 1-2 (1995)
92. Stefani, A., Strappavara, C.: Personalizing Access to Web Sites: The SiteIF Project. In: Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98 Pittsburgh, June 20-24 (1998) http://www.contrib.andrew.cmu.edu/~plb/HT98_workshop/Stefani/Stefani.html
93. Sugiyama, K., Hatano, K., Yoshikawa, M.: Adaptive web search based on user profile constructed without any effort from users. In: Proceedings 13 th International Conference on World Wide Web, New York, May 17-22 (2004) 675-684
94. SurfSaver, http://www.surfsaver.com/ (last access on October 2005)
95. Tan, A.: Adaptive Resonance Associative Map. Neural Networks, 8(3) (1995) 437-446
96. Tan, A., Teo, C.: Learning user profiles for personalized information dissemination. In: Proceedings of 1998 IEEE International Joint Conference on Neural Networks, Alaska, May 4-9 (1998) 183-188
97. Tanudjaja, F., Mui, L.: Persona: A Contextualized and Personalized Web Search. In: Proc 35th Hawaii International Conference on System Sciences, Big Island, Hawaii, January (2002) 53
98. Teevan, J., Dumais, S., Horvitz, E.: Personalizing Search via Automated Analysis of Interests and Activities. In: Proceedings of 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, August 15-19 (2005) 449-456
99. Trajkova, J., Gauch, S.: Improving Ontology-Based User Profiles. In: Proceedings of RIAO 2004, University of Avignon (Vaucluse), France, April 26-28 (2004) 380-389
100. Wærn, A.: User Involvement in Automatic Filtering: An Experimental Study. In User Modeling and User-Adaptive Interaction, 14 (2-3), June (2004) 201-237
101. Web-Ontology(WebOnt) Working Group, http://www.w3.org/2001/sw/WebOnt/ (last access on February 2004)
102. White, R.W., Jose, J.M., Ruthven, I.: Comparing explicit and implicit feedback techniques for Web retrieval: TREC-10 interactive track report. In: Proceedings of the Tenth Text Retrieval Conference (TREC 2001, Gaithersburg, MD) 534-538 http://trec.nist.gov/pubs/trec10/papers/glasgow.pdf
103. Widyantoro, D.H., Yin, J., El Nasr, M., Yang, L., Zacchi, A., Yen, J.: Alipes: A Swift

Messenger In Cyberspace. In: Proc. 1999 AAAI Spring Symposium Workshop on Intelligent Agents in Cyberspace, Stanford, March 22-24 (1999) 62-67 http://citeseer.ist.psu.edu/widyantoro99alipes.html
104. Widyantoro, D.H., Ioerger, T.R., Yen, J.: Learning User Interest Dynamics with Three-Descriptor Representation. Journal of the American Society of Information Science and Technology (JASIST) 52(3) February (2001) 212-225
105. The Wordnet Website, http://wordnet.princeton.edu/ (last access on February 2006)
106. eXtensible Markup Language, http://www.xml.com (last access on October 2005)
107. Yam search engine, http://www.yam.com (last access on October 2005)
108. Yan, T., García-Molina, H.: SIFT – A Tool for Wide-Area Information Dissemination. In: Proceedings of USENIX Technical Conference, New Orleans, Louisiana, January 16-20 (1995) 177-186
109. Yang, Y., Liu, X.: A Re-Examination Of Text Categorization Methods. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, California, August 15-19 (1999) 42-49
110. Yahoo Personalized Portal, http://my.yahoo.com/ (last access on September 2005)
111. Yahoo Directory, http://dir.yahoo.com/ (last access on October 2005)
112. Zhu, H., Zhong, J., Li, J., Yu, Y.: An Approach for Semantic Search by Matching RDF Graphs. In: Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, Pensacola Beach, Florida, May 14-16 (2002) 450-454