Intelligent User Profiling

Silvia Schiaffino[1,2] and Analía Amandi[1,2]

[1] ISISTAN Research Institute, Universidad Nacional del Centro de la Provincia de Buenos Aires, Campus Universitario, Argentina
[2] CONICET, Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina
{sschia,amandi}@exa.unicen.edu.ar

Abstract. User profiles or user models are vital in many areas in which it is essential to obtain knowledge about users of software applications. Examples of these areas are intelligent agents, adaptive systems, intelligent tutoring systems, recommender systems, intelligent e-commerce applications, and knowledge management systems. In this chapter we study the main issues regarding user profiles from the perspectives of these research fields. We examine what information constitutes a user profile; how the user profile is represented; how the user profile is acquired and built; and how the profile information is used. We also discuss some challenges and future trends in the intelligent user profiling area.

1 Introduction

A profile is a description of someone containing the most important or interesting facts about him or her. In the context of users of software applications, a user profile or user model contains essential information about an individual user. The motivation of building user profiles is that users differ in their preferences, interests, background and goals when using software applications. Discovering these differences is vital to providing users with personalized services.

The content of a user profile varies from one application domain to another. For example, if we consider an online newspaper domain, the user profile contains the types of news (topics) the user likes to read, the types of news (topics) the user does not like to read, the newspapers he usually reads, and the user's reading habits and patterns. In a calendar management domain the user profile contains information about the dates and times when the user usually schedules each type of activity in which he is involved, the priorities each activity feature has for the user, the relevance of each user contact and the user's scheduling and rescheduling habits. In other domains personal information about the user, such as name, age, job, and hobbies might be important.

Not only the content of user profiles differs from one domain to another, but also how the information they contain is acquired. The content of a user profile can be explicitly provided by the user or it has to be learned using some intelligent technique. User profiling implies inferring unobservable information about users from observable information about them, that is, their actions or utterances (Zukerman and Albrecht, 2001). A wide variety of Artificial Intelligence techniques have been used for user profiling, such as case-based reasoning (Lenz et al, 1998; Godoy et al., 2004), Bayesian networks (Horvitz et al, 1998; Conati et al, 2002; Schiaffino and Amandi, 2005; Garcia et al, 2007), association rules (Adomavicius and Tuzhilin, 2001; Schiaffino and Amandi, 2006), genetic algorithms (Moukas, 1996; Yannibelli et al, 2006), neural networks (Yasdi, 1999; Villaverde et al, 2006), among others.

The purpose of obtaining user profiles is also different in the various areas that use them. In adaptive systems, the user profile is used to provide the adaptation effect, that is to behave differently for different users (Brusilovsky and Millán, 2007). In intelligent agents, particularly in interface agents, the user profile is used to provide personalized assistance to users with respect to some software application (Maes, 1994). In intelligent tutoring systems, the user profile or student model is used to guide students in their learning process according to their knowledge and learning styles (Garcia et al, 2007). In e-commerce applications the user or customer profile is used to make personalized offers and to suggest or recommend products the user is supposed to like (Adomavicius and Tuzhilin, 2001). In knowledge management systems, the skills a user or employee has, the roles he takes within an organization, and his performance in these roles are used by managers or project leaders to assign him to the job position that suits him best (Sure et al, 2000). In recommender systems the user profile contains ratings for items like movies, news or books, which are used to recommend potentially interesting items to him and to other users with similar tastes or interests (Resnick and Varian, 1997).

In this Chapter we study user profiles from the different perspectives mentioned above. In Section 2 we describe what information constitutes a user profile. In Section 3 we examine the different ways in which we can acquire information about a user and then build a user profile. Section 4 focuses on intelligent user profiling techniques. Finally, Section 5 presents some future trends.

2 User Profile Contents

A user profile is a representation of information about an individual user that is essential for the (intelligent) application we are considering. This section describes the most common contents of user profiles: user interests; the user's knowledge, background and skills; the user's goals; user behaviour; the user's interaction preferences; the user's individual characteristics; and the user's context. We analyze and provide examples for the different contents in areas like intelligent agents, adaptive systems, intelligent tutoring systems, recommender systems, and knowledge management systems.

2.1 Interests

User interests are one of the most important (and typically the only) part of the user profile in information retrieval and filtering systems, recommender systems, some interface agents, and adaptive systems that are information-driven such as encyclopedias, museum guides, and news systems (Brusilovsky and Millán, 2007). Interests can represent news topics, web page topics, document topics, work-related topics or hobbies-related topics. Sometimes user interests are classified as short-term interests or long-term interests. The interest of users in football may be a short-term interest if the user reads or listens to news about this topic only during the World Cup, or a long-term interest if the user is always interested in this topic. For example, *NewsDude* (Billsus and Pazzani, 1999), an interface agent that learns about a user's interests in daily news stories, considers information about recent events as short-term interests, and a user's general preferences for news stories as long-term interests.

The most common representation of user interests are keyword-based models. In these models interests are represented by weighted vectors of keywords. Weights traditionally represent the relevance of the word for the user or within the topic. These representations are common in the Information Filtering and Information Retrieval areas. For example *Letizia* (Lieberman et al, 2001a), a browsing assistant, uses TF-IDF (term frequency/inverse document frequency) vectors to model user interests. In this technique the weight of each word is calculated by comparing the word frequency in a document against the word frequency in all the documents in a corpus (Salton and McGill, 1983). This technique is also used in *NewsDude* (Billsus and Pazzani, 1999), where news stories are converted to TF-IDF vectors.

A more powerful representation of user interests is through topic hierarchies (Godoy et al, 2004). Each node in the hierarchy represents a topic of interest for a user, which is defined by a set of representative words. This representation technique is important when we want to model not only general user interests such as sports or economy, but also the sub-topics of these interests that are relevant to a given user. For example, the user profile can indicate that a certain user is interested in documents talking about a famous football player and not in sports or football in general. An example of a topic hierarchy containing a user's interests is shown in Figure 1.

Often, a topic ontology is used as the reference to construct a user interest profile. An ontology is a conceptualization of a domain into a human-understandable, but machine-readable format consisting of entities, attributes, relationships, and axioms (Guarino and Giaretta 1995). For instance, in *Quickstep* (Middleton et al, 2004), the authors represent user profiles in terms of a research paper topic ontology. This recommender system was built to help researchers in a computer science laboratory setting, representing user profiling with a research topic ontology and using ontological inference to assist the profiling process. Similarly, in (Liang et al, 2007) students' interests within an e-learning system are determined using a topic ontology.

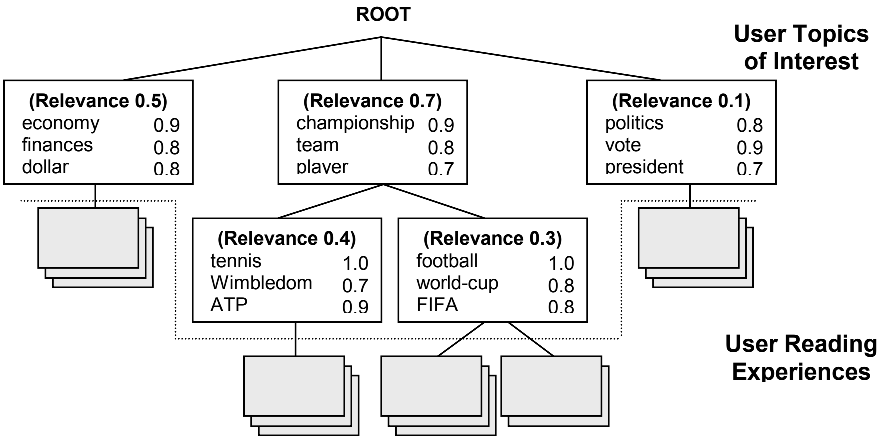


Fig. 1. Hierarchical representation of a user's interests

2.2 Knowledge, background and Skills

The knowledge the user has about the application domain, his background experience and his skills are important features within user profiles in different areas. In intelligent tutoring systems and adaptive educational systems, the student's knowledge about the subject taught is vital to provide proper assistance to the student or to adapt the content of courses according to it. This knowledge can be represented in different ways. The most common representation is through a model that keeps track of the student knowledge about every element in the course knowledge base. The idea is to mark each knowledge item X with a value calculated as "student knowledge of X". The value could be binary (knows - does not know), qualitative (good - average - bad) or quantitative, assigned as a probability of the student's familiarity with the item X. For instance, in Cumulate (Brusilovsky et al, 2005), the state of a student's knowledge is represented as a weighted overlay model covering a set of topics, and each educational activity can contribute to only one topic.

Another way of representing user's knowledge is through errors or misconceptions. In addition to (or instead of) modelling what the user knows, some works focus on modelling what the user does not know. For example, in (Chen and Hsieh 2005) the authors aim at diagnosing learners' common learning misconceptions during learning processes. They try to discover relationships between misconceptions.

Also, in many applications, the user's knowledge about the underlying domain is important. Some systems categorize users as expert, intermediate, or novice, depending on how well they know the application domain. For example, MetaDoc (Boyle and Encarnacion, 1994) considers the knowledge users have about Unix, which is the underlying application domain in this system.

Furthermore, user skills are key in areas like Knowledge Management. Within this area, skill management systems serve as technical platforms for mostly, though not exclusively, corporate-internal market places for skills and know-how. The systems are typically built on top of a database that contains profiles of employees and applicants. In this domain, profiles consist of numerous values for different skills and may be represented as vectors. In (Sure et al, 2000) authors use the integers "0" (no knowledge), "1" (beginner), "2" (intermediate) and "3" (expert) as skill values. Examples of skills can be "Programming in Y" or "Administration of Server X".

Finally, the user's background refers to those user's characteristics that are not directly related to the application domain. For instance, if we consider a tutoring system, the user's job or profession, his work experience, his traveling experience, the languages he speaks, among other information, constitute the user's background. As an application example, in (Cawsey et al, 2007) the authors describe an adaptive information system in the healthcare domain that considers users' literacy and medical background to provide them information that they can understand. The representation of users' background and skills is commonly done via stereotypes. We discuss them in Section 3.4.

2.3 Goals

Goals represent the user's objective or purpose with respect to the application he is working with, that is what the user wants to achieve. Goals are target tasks or subtasks at the focus of a user's attention (Horvitz et al, 1998). If the user is browsing the Web, his goal is obtaining relevant information (this type of goal is known as an information need). If the user is working with an e-learning system, his goal is learning a certain subject. In a calendar management system, the user's goals are scheduling new events or rescheduling conflicting events.

Determining what a user wants to do is not a trivial task. Plan recognition is a technique that aims at identifying the goal or intention of a user from the tasks he performs. In this context, a task corresponds to an action the user can perform in the software application, and a goal is a higher level intention of the user, which will be accomplished by carrying out a set of tasks. Systems using plan recognition observe the input tasks of a user and try to find all possible plans by which the observed tasks can be explained. These possible explanations or candidate plans are narrowed as the user continues performing further tasks. Plan recognition has been applied in different areas such as intelligent tutoring (Greer and Kohenn, 1995), interface agents (Lesh et al, 1999; Armentano and Amandi, 2006), and collaborative planning (Huber and Durfee, 1994).

Goals or intentions can be represented in different ways. Figure 2 shows a Bayesian network representation of a user's intentions in a calendar domain (Armentano and Amandi, 2006). In this representation, nodes represent user tasks and arcs represent probabilistic dependencies between tasks. Given evidence of a task performed by the user, the system can infer the next (most probable) task, and hence, the user's goal. Similarly, the *Lumiere* project at Microsoft Research (Horvitz et al., 1998) uses Bayesian networks to infer a user's needs by considering a user's background, actions and queries (help requests). Based on the beliefs of a user's needs and the utility theory of influence diagrams (an extension to Bayesian networks), an automated assistant provides help for users. In Andes (Gertner and VanLehn, 2000), plan recognition is necessary for the problem solving coach to select what step to suggest when a student asks for help. Since Andes wants to help students solve problems in their own way, it must determine what goal the student is probably trying to achieve, and suggest the action the student cannot perform due to lack of knowledge.

2.4 Behaviour

Usually, the user's behaviour with a software application is an important part of the user profile. If a given user behaviour is repetitive, then it represents a pattern that can be used by an adaptive system or an intelligent agent to adapt a web site or to assist the user according to the behaviour learnt. The type of behaviour modelled depends on the application domain. For example, *CAP (Calendar APprentice)* learns the scheduling behaviour of its user and learns rules that enable it to suggest the meeting duration, location, time, and date (Mitchell et al, 1994). In an intelligent e-commerce system, a behavioural profile models the customer's actions (Adomavicius and Tuzhilin, 2001). Examples of behaviours in this domain are "When purchasing cereal, John Doe usually buys milk" and "On weekends, John Doe usually spends more than $100 on groceries". In intelligent tutoring systems, the student behaviour is vital to assist him properly. In (Xu, 2002), a student profile is a set of <*t, e*> pairs, where *e* is a behaviour of the student and *t* expresses the time when the behaviour occurs. *t* could be a point in time or an interval of time. In this work, there are two main types of student behaviours, reading a particular topic and making a choice in a quiz.

Sometimes behaviours are routine, that is, they show some kind of regularity or seasonality. For example, QueryGuesser (Schiaffino and Amandi, 2005) models a user's routine queries to a database in a Laboratory Information Management System. In this agent, the user profile is composed of the queries each user performs

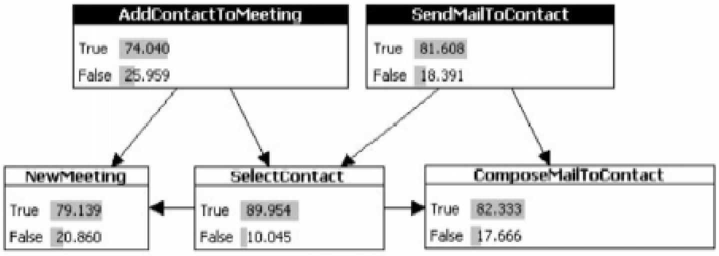


Fig. 2. Bayesian representation of a user's goals

and the moment when each query is generally made. The agent detects hourly, daily, weekly, and monthly behavioural patterns.

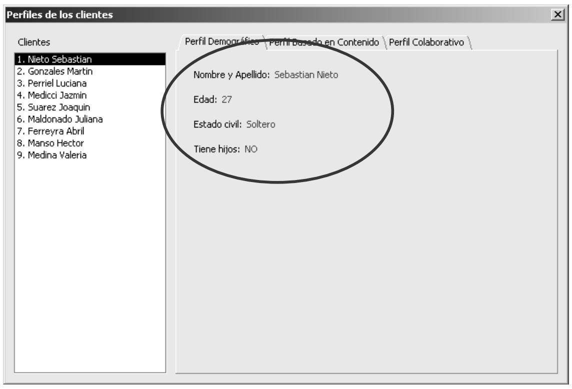
2.5 Interaction Preferences

A quite new component of a user profile is interaction preferences, that is, information about the user's interaction habits and preferences when he interacts with an interface agent (Schiaffino and Amandi, 2006). In interface agent technology, it is vital to know which agent's actions the user expects in different contexts and the modality of these actions. A user may prefer warnings, suggestions, or actions on the user's behalf. In addition, the agent can provide assistance by interrupting or not interrupting the user's work. A user interaction preference then expresses the preferred agent action and modality for different situations or contexts. As an illustration, consider an agent helping a user, John Smith, organize his calendar. Smith's current task is to schedule a meeting with several participants for the following Saturday in a free time slot. From past experience, the agent knows that one participant will disagree with the meeting date, because he never attends Saturday meetings. The agent can: warn the user about this problem, suggest another meeting date that considers all participant preferences and priorities, or do nothing. In this situation, some users would prefer a simple warning, while others would want suggestions about an alternative meeting date. In addition, when providing user assistance, agents can either interrupt the user's work or not. The agent must learn when the user prefers each modality. Information about these user preferences are kept in the user interaction profile, namely situations when the user: requires a suggestion to deal with a problem, needs only a warning about a problem, accepts an interruption from the agent, expects an action on his or her behalf, and wants a notification rather than an interruption.

2.6 Individual Characteristics

In some domains, personal information about the user is also part of the user profile. This item includes mainly demographic information such as gender, age, marital status, city, country, number of children, among other features. For example, Figure 3 shows the demographic profile of a customer in *Traveller*, a tourism recommender system that recommends package holidays and tours to customers.

On the other hand, a widely used user characteristic in intelligent tutoring systems and adaptive e-learning systems is the student's learning style. A learning-style model classifies students according to where they fit in a number of scales belonging to the ways in which they receive and process information. There have been proposed several models and frameworks for learning styles (Kolb 1984; Felder and Silverman, 1988; Honey and Mumford, 1992; Litzinger and Osif, 1993). For example, Felder and Silverman's model categorizes students as sensitive/intuitive, visual/verbal, active/reflective, and sequential/global, depending on how they learn. Various systems consider learning styles, such as ARTHUR (Gilbert



Clientes

1. Nieto Sebastian
2. Gonzales Martin
3. Perriel Luciana
4. Medicci Jazmin
5. Suarez Joaquin
6. Maldonado Juliana
7. Ferreyra Abril
8. Manso Hector
9. Medina Valeria

Nombre y Apellido: Sebastian Nieto

Edad: 27

Estado civil: Soltero

Tiene hijos: NO

Fig. 3. Demographic profile of a customer in *Traveller*

and Han, 1999) which models three learning styles (visual-interactive, reading-listener, textual), CS388 (Carver et al, 1996) and MAS-PLANG (Peña et al., 2002) that use Felder and Silverman styles; the INSPIRE system (Grigoriadou et al., 2001) that uses the styles proposed by Honey and Mumford.

Finally, personality traits are also important features in a user profile. A trait is a temporally stable, cross-situational individual difference. One of the most famous personality models is OCEAN (Goldberg, 1993). This model comprises five personality dimensions: Openness to Experience, Conscientiousness, Extraversion, Agreeableness, and Neuroticism. Personality models and the methods to determine personality are subjects widely studied in psychology (McCrae and Costa, 1996; Wiggins et al, 1988). In the area of user profiling, various methods are used to detect user's personality. For example, in (Arya et al, 2006) facial actions are used as visual cues for detecting personality.

2.7 Contextual Information

The user's context is a quite new feature in user profiling. There are several definitions of context, mostly depending on the application domain. According to (Dey and Abwod, 1999), context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. There are different types of contexts or contextual information that can be modelled within a user profile, as defined in (Goker and Myrhaug, 2002). The environmental context captures the entities that surround the user. These entities can, for instance, be things, services, temperature, light, humidity, noise, and persons. The personal context includes the physiological context and the mental context. The first part can contain information like pulse, blood pressure, weight, glucose level, retinal pattern, and hair colour. The latter part can contain information like mood, expertise, angriness, and stress. The social context describes the social aspects of the current user context. It can contain information about friends, neutrals, enemies, neighbours, co-workers, and relatives for instance. The spatio-temporal context describes aspects of the user context relating to the time and spatial extent for the user context. It can contain attributes like: time, location, or direction.

Context-aware systems (agents) are computing systems (agents) that provide relevant services and information to users based on their situational conditions or contexts (Dey and Abwod, 1999). In (Schiaffino and Amandi, 2006), for example, different types of assistance actions are executed by an agent depending on the task the user is carrying out and on the situation in which the user needs assistance. As regards users' emotions or mood, *RoCo* (Ahn and Picard, 2005) models different users' states, namely attentive, distracted, slumped, showing pleasure, showing displeasure, and acts accordingly. Other examples of context-aware systems based on the user location are various tourist guide projects where information is displayed depending on the current location of the user, such as (Yang et al, 1999).

2.8 Group Profiles

In contrast to individual user profiles, group profiles aim at combining individual user profiles to model a group. Group profiles are vital in those domains where it is necessary to make recommendations to groups of users rather than to individual users. Examples of these domains are tourism recommendation systems, movie recommenders, and adaptive television. In the first type of application, we find INTRIGUE (Ardissono et al, 2002), which recommends places to visit for tourist groups taking into account characteristics of subgroups within that group (such as children and disabled). Similarly, CATS (Collaborative Advisory Travel System) allows a group of users to simultaneously collaborate on choosing a skiing holiday package which satisfies the group as a whole (McCarthy et al, 2006). Group user feedback is used to suggest products that satisfy the individual and the group.

As regards TV, in (Masthoff, 2004) the authors discuss different strategies for combining individual user profiles to adapt to groups in an adaptive television application. In (Yu et al, 2006) the authors propose a recommendation scheme that merges individual user profiles to form a common user profile, and then generates common recommendations according to the common user profile.

3 Obtaining User Profiles

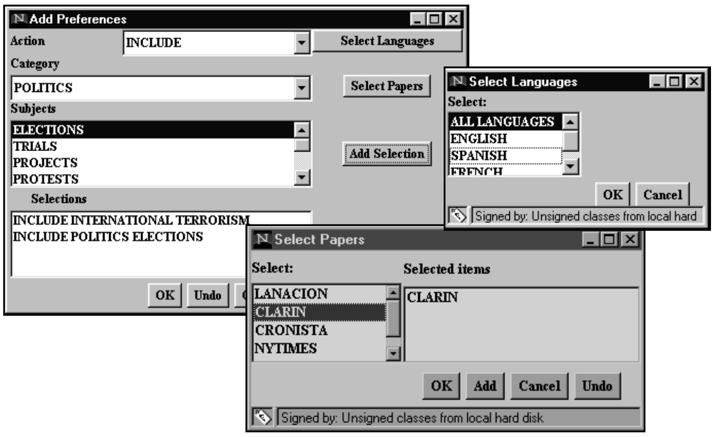
To build a user profile, the information needed can be obtained explicitly, that is provided directly by the user, or implicitly, through the observation of the user's actions. In this section we describe these alternatives.

3.1 Explicit Information

The simplest way of obtaining information about users is through the data they input via forms or other user interfaces provided for this purpose. Usually, this type of information is optional since users are not willing to fill in long forms providing information about them. Generally, the information gathered in this way is demographic, such as the user's age, gender, job, birthday, marital status, and hobbies. For eample, in (Adomavicius and Tuzhilin, 2001) this information constitutes the factual profile (name, gender, and date of birth), which is obtained by the e-commerce system from the customer's data.

In addition, personal interests can be informed explicitly. For example, in *NewsAgent* (Godoy et al, 2004) the user can indicate which sections of a digital newspaper he likes to read, which newspaper he prefers, or indicate general interesting topics, such as football, through a user interface, and he can also rate pages as interesting or uninteresting while he is reading. Figure 4 shows the user interfaces for these purposes. In *Syskill & Webert* (Pazzani et al, 1996), users make explicit relevance judgments of pages explored while browsing the Web. *Syskill & Webert* learns a profile from the user's ratings of pages and uses this profile to suggest other pages. The user can rate a page as either hot (two thumbs up), lukewarm (one thumb up and one thumb down), or cold (two thumbs down). The *Apt Decision* agent (Shearin and Lieberman, 2001) learns user preferences in the domain of rental real estate by observing the user's critique of apartment features. Users provide a small number of criteria in the initial interaction consisting of number of bedrooms, city, and price, then receive a display of sample apartments, and then react to any feature of any apartment independently, in any order.

Another way of providing explicit information is through the "Programming by Example" (PBE) or "Programming by Demonstration" paradigm (Lieberman, 2001b). In this approach, the user demonstrates examples to the computer. A



Action: INCLUDE

Category: POLITICS

Subjects: ELECTIONS, TRIALS, PROJECTS, PROTESTS

Selections:

INCLUDE INTERNATIONAL TERRORISM
INCLUDE POLITICS ELECTIONS

Select: ALL LANGUAGES, ENGLISH, SPANISH, FRENCH

Select: LANACION, CLARIN, CRONISTA, NYTIMES

Selected items: CLARIN

Signed by: Unsigned classes from local hard

Signed by: Unsigned classes from local hard disk

Fig. 4. Providing explicit information about a user's interests

software agent records the interactions between the user and a conventional interface, and writes a program that corresponds to the user's actions. The agent can then generalize the program so that it can work in other situations similar to, but not necessarily exactly the same as, the examples on which it is taught. For example, in (Ruvini and Dony, 2001) a software agent detects habitual patterns in a conventional programming language environment, Smalltalk, and automates those patterns.

3.2 Observation of a User's Actions

There are various problems with explicit user information. First, users are generally not willing to provide information by filling in long forms. Second, they not always tell or write the truth when completing forms about themselves. Third, although some of them might be willing to provide data, they sometimes do not know how to express their interests or what they really want. Thus, the most widely used method for obtaining information about users is observing their actions with the underlying application, recording or logging these actions, and discovering patterns from these logs through some Machine Learning or Data Mining technique.

In order to learn a user profile from a user's actions, there are certain conditions that must be fulfilled. The user behaviour has to be repetitive, that is the same actions have to be performed under similar conditions in different time points. If there is no repetition, no pattern can be discovered. In addition, the behaviour observed has to be different for different users. If not, there is no need for building an individual user profile.

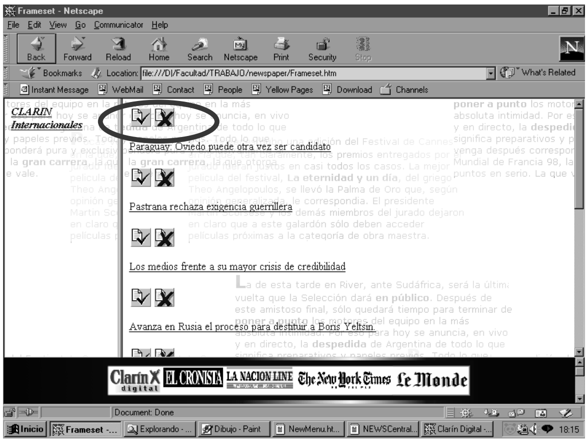
For example, *PersonalSearcher* (Godoy et al, 2004) unobtrusively observes a user's browsing behaviour in order to approximate the degree of user interest in each visited web page. In order to accomplish this goal, for each read page in a standard browser the agent observes a set of implicit indicators in a process known as implicit feedback (Oard and Kim, 1998). Implicit interest indicators used by *Personal Searcher* include the time consumed in reading a web page (considering its length), the amount of scrolling in a page, and whether it was added to the list of bookmarks or not. Similarly, *NewsAgent* monitors users' behaviour while they are reading newspapers on the web and it records information about the different articles they read and some indicators about their relevance to the user.

A key characteristic of learning through observation is that of adapting to the user's changing interests, preferences, habits and goals. The user profiling techniques used have to be able to adapt the content of the user profile as new observations are recorded. User feedback plays a fundamental role in this task, as explained in the next section.

3.3 User Feedback

User feedback is a key source of learning in interface agent technology. This feedback may be explicit, when users explicitly evaluate an agent's actions through a user interface provided for that purpose, or implicit, when the agent observes a user's actions after assisting him to detect some implicit evaluation of its assistance. The explicit feedback can be simple or complex. It is simple when the user is required to evaluate the agent's assistance according to a quantitative or a qualitative scale (for example 0 to 10 or relevant or irrelevant) or to just press a dislike/like button. However, it becomes more complicated when the user is required to provide big amounts of information in various steps. Mostly, an interface agent has to learn from implicit feedback since the explicit feedback is not always available. As said before, the reason is that not all users are willing to provide explicit feedback, mainly if this demands of them a lot of time and effort. For example, *NewsDude* (Billsus and Pazzani, 1999) supports the following feedback options: interesting, not interesting, I already know this, tell me more, and explain. In *NewsAgent* (Godoy et al, 2004), once the agent shows the personalized newspaper to the user, he can rate each news item contained in the newspaper as interesting or uninteresting, as shown in Figure 5.

User ratings in recommender systems can be considered also as user feedback (or as explicit information as well). In these systems, users rate items such as movies they have seen or books they have read. These ratings are used both to build the user profile and to recommend potentially interesting items to other users similar to the user under consideration. This last type of recommendation is known as collaborative recommendation or collaborative filtering. For example, *MovieLens*[1] uses user ratings to generate personalized recommendations for other movies the user will like and dislike.


Frameset - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks Location: file:///D|/Facultad/TRABAJO/newspaper/Frameset.htm
What's Related
Instant Message WebMail Contact People Yellow Pages Download Channels
CLARIN Internacionales
Clarín digital EL CRONISTA LA NACION LINE The New York Times Le Monde
Document: Done
Inicio Frameset -... Explorando - ... Dibujo - Paint NewMenu.ht... NEWSCentral... Clarín Digital -...
18:15

Paraguay: Oviedo puede otra vez ser candidato

Pastrana rechaza exigencia guerrillera

Los medios frente a su mayor crisis de credibilidad

Avanza en Rusia el proceso para destituir a Boris Yeltsin.

Fig. 5. Providing user feedback in *NewsAgent*

[1] http://movielens.umn.edu/

3.4 Stereotypes

A stereotype is the representation of relevant common characteristics of users pertaining to specific user subgroups of an application system (Kobsa, 2001). Stereotypes were the first attempt to differentiate a user from other users (Rich, 1979; Rich, 1989). Often, different system functionality was provided to users depending on their stereotype. The most popular stereotypes are: novice, intermediate or expert user. For example, UMT (Brajnik and Tasso, 1994) allows the user model developer the definition of hierarchically ordered user stereotypes; and BGP-MS (Kobsa and Pohl, 1995) allows assumptions about the user and stereotypical assumptions about user groups to be represented in a first-order predicate logic. Other examples of user stereotypes were presented in Section 2.2.

Stereotypes are useful when no other information about a user is available, that is, when the user has not used the system yet. This is the idea of um Toolkit (Kay, 1990), where information about a user stereotype is used as default information. Stereotypes enable the classification of users as belonging to one or more of a set of subgroups, and also the integration of the typical characteristics of these subgroups into the individual user profile.

4 Intelligent User Profiling Techniques

Intelligent user profiling implies the application of intelligent techniques, coming from the areas of Machine Learning, Data Mining or Information Retrieval, for example, to build user profiles. The data these techniques use to automatically build user profiles are obtained mainly from the observation of a user's actions, as described in the previous section. In this section we briefly describe three techniques widely used for user profiling and we present examples of their use that were developed in our research group.

4.1 Bayesian Networks

In the last decade, interest has been growing steadily in the application of Bayesian representations and inference methods for modelling the goals, preferences, and needs of users (Horvitz et al, 1998). A Bayesian network (BN) is a compact, expressive representation of uncertain relationships among variables of interest in a domain. A BN is a directed acyclic graph where nodes represent random variables and arcs represent probabilistic correlations between variables (Jensen, 2001). The absence of edges in a BN denotes statements of independence. A BN also represents a particular probability distribution, the joint distribution over all the variables represented by nodes in the graph. This distribution is specified by a set of conditional probability tables (CPT). Each node has an associated CPT that specifies the probability of each possible state of the node given each possible combination of states of its parents. For nodes without parents, probabilities are not conditioned on other nodes; these are called the prior or marginal probabilities of these variables.

The mathematical model underlying BN is Bayes' theorem, which is shown in Equation 1. Bayes' theorem relates conditional and marginal probabilities. It yields the conditional probability distribution of a random variable A, assuming we know: information about another variable B in terms of the conditional probability distribution of B given A, and the marginal probability distribution of A alone. Equation 1 reads: the probability of A given B equals the probability of B given A times the probability of A, divided by the probability of B.

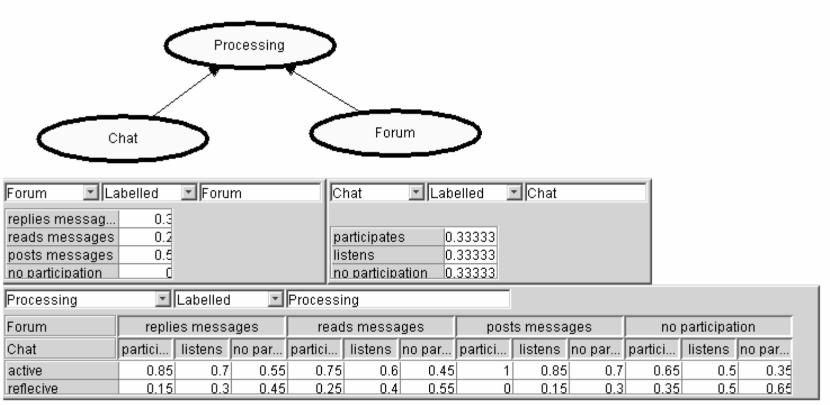
An important characteristic of BN is that Bayesian inference mechanisms can be easily applied to them. The goal of inference is typically to find the conditional distribution of a subset of the variables, conditioned on known values for some other subset (the evidence). Thus, a BN can be considered a mechanism for automatically constructing extensions of Bayes' theorem to more complex problems.

$$P(A/B) =(P(B/A)\, P(A)) / P(B) \tag{1}$$

As an example of using BN for user profiling, consider the work presented in (Garcia et al, 2007), where BN are used to model a student's behaviour with an e-learning system and to detect his learning style. In this example, random variables represent the different dimensions of Felder's learning styles (Felder and Silverman, 1988) and the factors that determine each of these dimensions. The dimensions modelled are perception, processing and understanding. The values these variables can take are sensory/intuitive, active/reflective, and sequential/global respectively. The factors that determine them are extracted from the interactions between the student and a web-based education system. Thus, a BN models the relationships between the dimensions of learning styles and the factors determining them. A part of the BN proposed in this work is shown in Figure 6. This network models the relationships between the participation of a student in chats and forums and the processing style of this student. Thus, the BN has three nodes: chat, forum, and processing. The "chat" node has three possible states: participates, listens, no participation. The "forum" node has four possible states: replies messages, reads messages, posts messages and no participation. Finally, the "processing" node has two possible values, namely active and reflective.

The Bayesian model is completed with the simple probability tables for the independent nodes and the CPT for the dependent nodes. The values of the simple probabilities are obtained by analyzing a student's log file. The probability functions associated with the independent nodes are gradually obtained by observing the student interaction with the system. The values of the CPT are set by combining expertise knowledge and experimental results. As an example, Figure 6 shows the probability values obtained for a certain student for the "chat", "forum" and "processing" nodes. We can observe the marginal probabilities for the independent nodes, namely chat and forum, and the CPT for the processing node.

Once the BN is built, the student learning style is determined via Bayesian inference. The authors infer the values of the nodes corresponding to the dimensions



Forum ▾ Labelled ▾ Forum

replies messag...	0.3
reads messages	0.2
posts messages	0.5
no participation	0

Chat ▾ Labelled ▾ Chat

participates	0.33333
listens	0.33333
no participation	0.33333

Processing ▾ Labelled ▾ Processing

Forum	replies messages			reads messages			posts messages			no participation		
Chat	partici...	listens	no par...	partici...	listens	no par...	partici...	listens	no par...	partici...	listens	no par...
active	0.85	0.7	0.55	0.75	0.6	0.45	1	0.85	0.7	0.65	0.5	0.35
reflecive	0.15	0.3	0.45	0.25	0.4	0.55	0	0.15	0.3	0.35	0.5	0.65

Fig. 6. Building a student profile with a BN

of a learning style given evidence of the student's behaviour with the system. The learning style of the student is the one having the greatest posterior probability value. In the simple example in Figure 6, given evidence of the utilization of the chat and forum facilities, we could infer whether the student processes information actively (discussing, in groups) or reflectively (by himself).

There are various works that use BN for user profiling. For example, the Lumiere project at Microsoft Research (Horvitz et al., 1998) uses BN to infer a user's needs by considering a user's background, actions and queries. Based on the beliefs of a user's needs an automated assistant provides help for users. In (Sanguesa et al, 1998) the authors use BN to model the profile of a web visitor and they use this profile to recommend interesting web pages. ANDES (Gertner and VanLehn, 2000) and SE-Coach (Conati and VanLehn, 2000) use this technique to model student knowledge in Physics. In (Gamboa and Fred, 2001) the authors use BN to assess students' state of knowledge and learning preferences in an intelligent tutoring system.

4.2 Association Rules

Association rules are a data mining technique widely used to discover patterns from data. They have also been used to learn user profiles in different areas, mainly in those related to e-commerce (Adomavicius and Tuzhilin, 2001) and web usage (Gery and Hadad, 2003). An association rule is a rule which implies certain association relationships among a set of objects in a given domain, such as they occur together or one implies the other. Association rule mining is commonly stated as follows (Agrawal and Srikant, 1994): Let *I* be a set of items and *D* be a set of transactions, each consisting of a subset *X* of items in *I*. An association rule is an implication of the form X→Y, where $X \subset I$, $Y \subset I$ and $X \cap Y = \emptyset$. *X* is the antecedent of the rule and *Y* is the consequent. The rule has support *s* in *D* if *s*% of the transactions in *D* contains $X \cap Y$. The rule X→Y holds in *D* with confidence *c* if *c*% of transactions in *D* that contain *X* also contain *Y*. Given a transaction database *D*, the problem of mining association rules is to find all association rules that satisfy: minimum support (called *minsup*) and minimum confidence (called *minconf*).

For example, in (Schiaffino and Amandi, 2006) association rules are used to discover a user's interaction preferences with an interface agent. Different algorithms have been proposed for association rule mining. In this work, authors use the Apriori algorithm (Agrawal and Srikant, 1994) to generate association rules from a set of user-agent interaction experiences. An interaction experience describes a unique interaction between the user and the agent, which can be initiated by any of them. The interaction records the situation or context originating it, the assistance the agent provided, the task the user was carrying out when the interaction took place, the modality of the assistance, the user feedback to the assistance type and the modality (if available), and an evaluation of the interaction (success, failure, or undefined).

Association rule mining algorithms tend to produce huge amounts of rules, most of them irrelevant or uninteresting. Therefore, some post-processing steps are needed to obtain valuable information to build a user profile. In the example we are describing, the rules generated by Apriori are automatically post-processed in order to derive useful knowledge about the user from them. Post-processing includes detecting the most interesting rules, eliminating redundant and insignificant rules, eliminating contradictory weak rules, and summarizing the information in order to formulate hypotheses about a user's preferences more easily. For filtering rules, the authors use templates to express and select relevant rules. For example, they are interested in those association rules of the form "*situation, assistance action→user feedback, evaluation*"; and also in association rules of the form "*situation, modality, [user task], [relevance], [assistance action] → user feedback, evaluation*", where brackets mean that the attributes are optional. To eliminate redundant rules, they use a subset of the pruning rules proposed in (Shah, 1999). Basically, these pruning rules state that given the rules A,B→C and A→C, the first rule is redundant because it gives little extra information. Thus, it can be deleted if the two rules have similar confidence values. Similarly, given the rules A→B and A→B,C, the first rule is redundant since the second consequent is more specific. Thus, the redundant rule can be deleted provided that both rules have similar confidence values. A contradictory rule is one indicating a different assistance action (modality) for the same situation, and having a small confidence value with respect to the rule being compared. After pruning, rules are grouped by similarity and a hypothesis is generated considering: a main rule, positive evidence (redundant rules that could not be eliminated), and negative evidence (contradictory rules not eliminated). Once a hypothesis is formulated, the profiling algorithm computes the certainty degree of the hypothesis by taking into account the support values of the main rule, the positive and the negative evidence. The whole user profiling process is summarized in Figure 7.

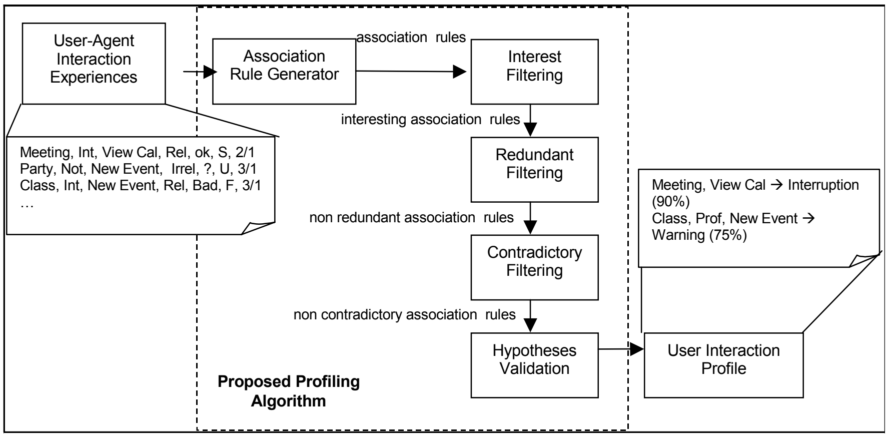

User-Agent Interaction Experiences
Association Rule Generator
association rules
Interest Filtering
interesting association rules
Redundant Filtering
non redundant association rules
Contradictory Filtering
non contradictory association rules
Hypotheses Validation
User Interaction Profile
Meeting, Int, View Cal, Rel, ok, S, 2/1
Party, Not, New Event, Irrel, ?, U, 3/1
Class, Int, New Event, Rel, Bad, F, 3/1
…
Meeting, View Cal → Interruption (90%)
Class, Prof, New Event → Warning (75%)
Proposed Profiling Algorithm

Fig. 7. User profiling with association rules

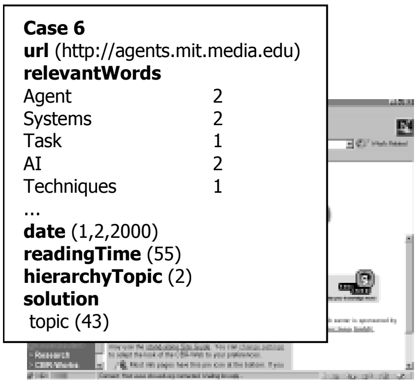
Other works have utilized association rules for building user profiles. In (Chen and Hsieh 2005) the authors use this technique for diagnosing learners' common learning misconceptions during learning processes. In this work, the association rules state that if misconception A occurs then misconception B is probable to occur. In (Adomavicius and Tuzhilin, 2001) association rules are used to build customers' profiles in an e-commerce application. Association rules indicate relationships among items bought by a customer.

4.3 Case-Based Reasoning

CBR is a technique that solves new problems by remembering previous similar experiences (Kolodner, 1993). A case-based reasoner represents problem-solving situations as cases. Given a new situation, it retrieves relevant cases (the ones matching the current problem) and it adapts their solutions to solve the problem. In an interpretative approach, CBR is applied to accomplish a classification task, that is, find the correct class for an unclassified case. The class of the most similar past case becomes the solution to the classification problem.

CBR has been used to build user profiles in areas like information retrieval and information filtering (Lenz et al, 1998; Smyth and Cotter, 1999). For example, in (Godoy et al, 2004) CBR is used to obtain a user interest profile. In this domain, cases represent relevant documents read by a user on the web. Each case records the main characteristics of a document, which enable the reasoner to determine its topic considering previous readings. Document topics represent a user's interests and they constitute case solutions. A topic or category is extensionally defined by a set of cases sharing the same solution. Using this approach, previously read documents can help to categorize new ones into specific categories, assuming that similar documents share the same topic.

Cases represent specific and contextual knowledge that describes a particular situation. In the example we are describing, cases represent readings of web documents relevant to a user. A case has three main parts: the description of the situation or problem, the solution, and the outcome or results of applying the solution to the problem. In this example, the description of the situation includes the URL (Uniform Resource Locator) of the document, a general category the document belongs to, a vector of representative words, and the time the user spent reading it. The solution is a code number that identifies a certain topic of interest for a user and the outcome describes in some way the user's feedback. The most important part of a document representation as a case is the list of relevant words. The authors use a bag-of-words representation where each word in the document has an associated weight that depends on the frequency of a word in the document and an additive factor defined in terms of several word characteristics in the document. The additive factor is calculated taking into account the word location inside the HTML document structure (words in the title are more important than words in the document body) and the word style (bold, italic or underlined). An example of a case in this domain is shown in Figure 8.



Case 6
url (http://agents.mit.media.edu)
relevantWords

Agent	2
Systems	2
Task	1
AI	2
Techniques	1
...	

date (1,2,2000)
readingTime (55)
hierarchyTopic (2)
solution
topic (43)

Fig. 8. Case representing an interesting web page

The comparison of cases is performed through a number of dimensions that describe them. A similarity function is defined for each of these dimensions, the most important being the one that measures the similarity between the relevant word lists. This similarity is calculated by the inner product with cosine normalization (Salton and McGill, 1983). A numerical evaluation function that combines the matching of each dimension with the importance value assigned to that dimension, is used to obtain the global similarity between the entry case C_E and the retrieved one C_R. The function used is the formula in Equation 2, where w_i is the importance of the dimension i given by the agent designer; sim_i is a similarity function for this dimension; f_i^E, f_i^R are the values for the feature f_i in both cases. If the similarity value obtained is higher than a given threshold, the cases are considered similar, and then, we can conclude that both cases are in the same user interest topic.

$$S(C_E,C_R)=\Sigma^n_{i=1}\, w_i * sim_i\, (f_i^E, f_i^R) \quad (2)$$

4.4 Other User Profiling Techniques

Many other Machine Learning techniques have been used for user profiling, such as genetic algorithms, neural networks, kNN-algorithm, clustering, and classification techniques such as decision trees or naïve Bayes classifier. For example, *Personal WebWatcher* (Mladenic, 1996) and *Syskill&Webert* (Pazzani et al, 1996) use naive Bayes classifiers for detecting users' interests when browsing the web. *Amalthaea* (Moukas, 1996) uses genetic algorithms to evolve a population of vectors representing a user's interests. The user profile is used to discover and filter information according to the user's interests. *NewsDude* (Billsus and Pazzani, 1999) obtains a short-term interest user profile using the k-NN algorithm and a long-term interest profile using a naïve Bayes classifier. *PersonalSearcher* (Godoy and Amandi, 2006) uses a clustering algorithm to categorize web documents and hence determine a user's interest profile. *SwiftFile* uses a TF-IDF style classifier to organize emails (Segal and Kephart, 2000). *CAP* uses decision trees to learn users' scheduling preferences (Mitchell et al., 1994).

Combinations of different techniques have also been used for building user profiles. For example, in (Martin-Bautista et al, 2000) the authors combine genetic algorithms and classification techniques (fuzzy logic) to build user profiles from a collection of documents previously retrieved by the user. In (Schiaffino and Amandi, 2000) case-based reasoning and Bayesian networks are combined to learn a user profile in a LIMS (Laboratory Information Management System). The user profile comprises routine user queries that represent a user's interests in the LIMS domain. In (Ko and Lee,2000) the authors combine genetic algorithms and a naive Bayes classifier to recommend interesting web documents to users .

5 Future Trends

We have studied in this Chapter the main issues concerning user profiles: how a user profile is composed; how a user profile can be acquired; and how a user profile can be used. We have seen that user profiles are vital in many areas; many of them in constant evolution and some new ones. Thus, researchers in the area of user profiling have to fulfill the expectations of these new trends and include new components as part of a profile and develop new techniques to build them.

As regards user profile contents, in recent years there has been increasing interest in modelling users' emotions and moods as part of the user profiles in areas such as social computing and intelligent agents. Emotional state has a similar structure to personality (described in Section 2.6), but it changes over time. The emotional state is a set of emotions that have a certain intensity. For example, the OCC model (Ortony et al, 1988) defines 22 emotions. An example of work in this direction is *AutoTutor* (D'Mello et al, 2007), which tries to determine students' emotions as they interact with an intelligent tutoring system. It uses several non-intrusive sensing devices to obtain this information. *AutoTutor* analyzes facial expressions, posture patterns, and conversational cues to determine a student's emotional state.

With respect to contextual information about a user, the developments in the areas of ubiquitous computing, mobile devices, and physical sensors enable the incorporation of new features in user profiles such as the focus of user attention (detected via eye-tracking), users' mood and emotions (detected analyzing facial expressions and body posture), temperature and humidity of the user's location, among others.

The area of Knowledge Management is acquiring great interest nowadays for organizations of different types. Within this area, user profiling is vital for different purposes. For example, building an employee profile focused on the employee's skills is important to place him in the position that best suits him. This is the purpose of skills management systems. Also, building a customer profile is important for customer relationship management (CRM). For example, in a credit card company, information such as the type of products the customer usually buys, how much he spends, when and where he buys what product, and how his family is composed, is key to offering him a personalized service.

Building group profiles is also a new tendency. Some works are being carried out in this direction (Jameson and Smyth, 2007). The challenges in this area are how to combine individual preferences into a group profile, how to help users to reach some kind of consensus, and how to make group recommendations trying to maximize average satisfaction, minimize misery and/or ensure some degree of fairness among participants.

References

Adomavicius, G., Tuzhilin, A.: Using Data Mining Methods to Build Customer Profiles. IEEE Computer 34(2) (2001)

Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proc. of the 20th Int'l Conference on Very Large Databases, Chile (1994)

Ahn, H., Picard, R.: Affective Cognitive Learning and Decision Making: A Motivational Reward Framework For Affective Agents. In: The 1st International Conference on Affective Computing and Intelligent Interaction, Beijing, China (2005)

Ardissono, L., Goy, A., Petrone, G., Segnan, M., Torasso, P.: Ubiquitous User Assistance in a Tourist Information Server. In: De Bra, P., Brusilovsky, P., Conejo, R. (eds.) AH 2002. LNCS, vol. 2347, pp. 14–23. Springer, Heidelberg (2002)

Armentano, M., Amandi, A.: A Bayesian Networks Approach to Plan Recognition for Interface Agents. In: Proc. Argentine Symposium on Artificial Intelligence, pp. 1–12 (2006)

Arya, A., Jefferies, N., Enns, J., DiPaola, S.: Facial actions as visual cues for personality. Computer Animation and Virtual Worlds 17(3-4), 371–382 (2006)

Baghaei, N., Mitrovic, A.: Evaluating a Collaborative Constraint-based Tutor for UML Class Diagrams. Frontiers in Artificial Intelligence and Applications, vol. 158. IOS Press, Amsterdam (2007)

Billsus, D., Pazzani, M.: A Personal News Agent that Talks, Learns and Explains. In: Proc. 3rd Int. Conf. on Autonomous Agents (Agents 99), Seattle, Washington (1999)

Boyle, C., Encarnacion, A.: MetaDoc: an adaptive hypertext reading system. User Modeling and User-adapted Interaction 4, 1–19 (1994)

Brajnik, G., Tasso, C.: A shell for developing non-monotonic user modeling systems. International Journal of Human-Computer Studies 40, 31–62 (1994)

Brusilovsky, P., Sosnovsky, S., Shcherbinina, O.: User Modeling in a Distributed E-learning Architecture. In: Ardissono, L., Brna, P., Mitrović, A. (eds.) UM 2005. LNCS (LNAI), vol. 3538, pp. 387–391. Springer, Heidelberg (2005)

Brusilovsky, P., Millán, E.: User Models for Adaptive Hypermedia and Adaptive Educational Systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 3–53. Springer, Heidelberg (2007)

Carver, C.A., Howard, R.A., Lavelle, E.: Enhancing student learning by incorporating learning styles into adaptive hypermedia. In: Proceedings of 1996 ED-MEDIA World Conf. on Educational Multimedia and Hypermedia, Boston, USA, pp. 118–123 (1996)

Cawsey, A., Grasso, F., Paris, C.: Adaptive Information for Consumers of Healthcare. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 465–484. Springer, Heidelberg (2007)

Chen, C., Hsieh, Y.: Mining Learner Profile Utilizing Association Rule for Common Learning Misconception Diagnosis. In: ICALT 2005, pp. 588–592 (2005)

Conati, C., VanLehn, K.: Toward computer-based support of meta-cognitive skills: A computational framework to coach self-explanation. The International Journal of Artificial Intelligence in Education 11, 389–415 (2000)

Conati, C., Gertner, A., VanLehn, K.: Using Bayesian Networks to Manage Uncertainty in Student Modeling. User Modeling and User-Adapted Interaction 12(4), 371–417 (2002)

Dey, A., Abwod, G.: Towards a better understanding of context and context-awareness. GVU Technical Report GIT-GVU-99-22 (1999), Also In the Workshop on The What, Who, Where, When, and How of Context-Awareness, CHI 2000

D'Mello, S.K., Picard, R.W., Graesser, A.C.: Towards an Affect-Sensitive AutoTutor. IEEE Intelligent Systems (Special issue on Intelligent Educational Systems) 22(4), 53–61 (2007)

Felder, R., Silverman, L.: Learning and Teaching Styles in Engineering Education. Engineering Education 78(7), 674–681 (1988)

Gamboa, H., Fred, A.: Designing intelligent tutoring systems: a bayesian approach. In: ICEIS Artificial Intelligence and Decision Support Systems, pp. 452–458 (2001)

Garcia, P., Amandi, A., Schiaffino, S., Campo, M.: Evaluating Bayesian Networks' Precision for Detecting Students' Learning Styles. Computers and Education 49(3), 794–808 (2007)

Gertner, A.S., VanLehn, K.: Andes: A Coached Problem Solving Environment for Physics. In: Gauthier, G., VanLehn, K., Frasson, C. (eds.) ITS 2000. LNCS, vol. 1839, pp. 133–148. Springer, Heidelberg (2000)

Gery, M., Hadad, H.: Evaluation of web usage mining approaches for user's next request prediction. In: Proceedings of the 5th ACM international workshop on Web information and data management, pp. 74–81 (2003)

Gilbert, J., Han, C.: Arthur: An Adaptive Instruction System Based on Learning Styles. In: Proceedings of International Conference on Mathematics / Science Education and Technology, pp. 100–105 (1999)

Godoy, D., Schiaffino, S., Amandi, A.: Interface Agents Personalizing Web-based Tasks. Cognitive Systems Research Journal (Special Issue on Intelligent Agents and Data Mining for Cognitive Systems) 5, 207–222 (2004)

Godoy, D., Amandi, A.: A Conceptual Clustering Approach for User Profiling in Personal Information Agents. AI Communications 19(3), 207–227 (2006)

Goker, A., Myrhaug, H.I.: User context and personalization. In: Proceedings of ECCBR Workshop on Case Based Reasoning and Personalization, UK (2002)

Goldberg, L.R.: The structure of phenotypic personality traits. American Psychologist 48, 26–34 (1993)

Greer, J., Koehn, G.: The peculiarities of plan recognition for intelligent tutoring systems. In: Proceedings of the workshop on The Next Generation of Plan Recognition Systems: Challenges for and Insight from Related Areas of AI, pp. 54–59 (1995)

Grigoriadou, M., Papanikolaou, K., Kornilakis, H., Magoulas, G.: INSPIRE: an intelligent system for personalized instruction in a remote environment. In: Proceedings of 3rd Workshop on Adaptive Hypertext and Hypermedia, pp. 13–24 (2001)

Guarnino, N., Giaretta, P.: Ontologies and knowledge bases: Towards a terminological clarification. In: Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)

Honey, P., Mumford, A.: The Manual of Learning Styles. Maidenhead (1992)

Horvitz, E., Breese, J., Heckerman, D., Hovel, D., Rommelse, K.: The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 256–265 (1998)

Huber, M., Durfee, E., Wellman, M.: The automated mapping of plans for plan recognition. In: Workshop on Distributed Artificial Intelligence, pp. 137–152 (1994)

Jameson, A., Smyth, B.: Recommendation to groups. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 596–627. Springer, Heidelberg (2007)

Jensen, F.: Bayesian Networks and Decision Graphs. Springer, Heidelberg (2001)

Kay, J.: um: a user modeling toolkit. In: Proc. 2nd International User Modeling Workshop, Hawaii, p. 11 (1990)

Ko, S.J., Lee, J.H.: Discovery of User Preference through Genetic Algorithm and Bayesian Categorization for Recommendation. In: Arisawa, H., Kambayashi, Y., Kumar, V., Mayr, H.C., Hunt, I. (eds.) ER Workshops 2001. LNCS, vol. 2465, pp. 471–484. Springer, Heidelberg (2002)

Kobsa, A., Pohl, W.: The BGP-MS user modeling system. User Modeling and User-Adapted Interaction 4(2), 59–106 (1995)

Kobsa, A.: Generic User Modeling Systems. User Modeling and User Adapted Interaction 11, 49–63 (2001)

Kolb, D.A.: Experiential learning: Experience as the source of learning and development. Prentice Hall, Upper Saddle River (1984)

Kolodner, J.: Case-based reasoning. Morgan Kaufmann, San Francisco (1993)

Lenz, M., Hubner, A., Kunze, M.: Question Answering with Textual CBR. In: Proceedings of the International Conference on Flexible Query Answering Systems, Denmark (1998)

Lesh, N.B., Rich, C., Sidner, C.L.: Using Plan Recognition in Human-Computer Collaboration. In: International Conference on User Modeling, June 1999, pp. 23–32 (1999)

Liang, Y., Zhao, Z., Zeng, Q.: Mining Users' Interests from Reading Behaviour in E-learning Systems. In: 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 417–422 (2007)

Lieberman, H., Fry, C., Weitzman, L.: Exploring the Web with Reconnaissance Agents. Communications of the ACM, 69–75 (Aug. 2001a)

Lieberman, H. (ed.): Your wish is my command: Programming by Example. Morgan Kaufman, San Francisco (2001b)

Litzinger, M.E., Osif, B.: Accommodating diverse learning styles: Designing instruction for electronic information sources. In: Shirato, L. (ed.) What is GoodInstruction Now? Library Instruction for the 90s, Pierian Press, Ann Arbor (1993)

Maes, P.: Agents that reduce work and information overload. Communications of the ACM 37(7), 31–40 (1994)

Martin-Bautista, M.J., Vila, M.A., Larsen, H.L.: Building adaptive user profiles by a genetic fuzzy classifier with feature selection. In: The Ninth IEEE International Conference on Fuzzy Systems (2000)

Masthoff, J.: Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers. User Modeling and User Adapted Interaction 14, 35–87 (2004)

McCarthy, K., Salamó, M., Coyle, L., McGinty, L., Smyth, B., Nixon, P.: Group Recommender Systems: A critiquing based approach. In: Proc. Intelligent User Interfaces, IUI 06 (2006)

McCrae, R., Costa Jr., P.T.: Toward a new generation of personality theories: Theoretical contexts for the five-factor model. In: Wiggins, J.S. (ed.) The five-factor model of personality: Theoretical perspectives, pp. 51–87. Guilford, New York (1996)

Middleton, S.E., Shadbolt, N.R., Roure, D.C.: Ontological user profiling in recommender systems. ACM Transactions on Information Systems (TOIS) 22(1), 54–88 (2004)

Mitchell, T., Caruana, R., Freitag, D., McDermoot, J., Zabowski, D.: Experience with a learning Personal Assistant. Communications of the ACM 37(7), 81–91 (1994)

Mladenic, D.: Personal WebWatcher: Implementation and Design. Technical Report IJS-DP-7472, Department of Intelligent Systems, J. Stefan Institute, Slovenia (1996)

Moukas, A.: Amalthaea: Information Discovery and Filtering using a Multi-agent Evolving Ecosystem. In: Proceedings of the Conference on the Practical Application of Intelligent Agents and MultiAgent Technology, London, UK (1996)

Oard, D., Kim, J.: Implicit feedback for recommender systems. In: Proceedings of the AAAI Workshop on Recommender Systems (1998)

Ortony, A., Clore, G.L., Collins, A.: The Cognitive Structure of Emotions. Cambridge University Press, Cambridge (1988)

Pazzani, M., Muramatsu, J., Billsus, D.: Syskill & Webert: Identifying Interesting Web Sites. AAAI/IAAI, vol. 1, pp. 54–61 (1996)

Peña, C., Marzo, J., de la Rosa, J.: Intelligent agents in a teaching and learning environment on the Web. In: Proceedings ICALT 2002, Rusia (2002)

Resnick, P., Varian, H.: Recommender Systems. Communications of the ACM 40(3), 56–58 (1997)

Rich, E.: User modeling via stereotypes. Cognitive Science 3, 355–366 (1979)

Rich, E.: Stereotypes and user modeling. In: Kobsa, A., Wahlster, W. (eds.) User Models in Dialog Systems, pp. 35–51. Springer, Heidelberg (1989)

Ruvini, J.D., Dony, C.: Learning Users' Habits to Automate Repetitive Tasks. In: Your wish is my command: Programming by Example, Morgan Kaufman, San Francisco (2001)

Salton, G.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)

Sanguesa, R., Cortés, U., Nicolás, M.: BayesProfile: application of Bayesian Networks to website user tracking. Technical Report, Universidad de Catalonia (1998)

Segal, R., Kephart, J.: Swiftfile: An intelligent assistant for organizing e-mail. In: AAAI 2000 Spring Symposium on Adaptive User Interfaces, Stanford, CA (2000)

Schiaffino, S., Amandi, A.: User Profiling with Case-Based Reasoning and Bayesian Networks. In: Open Discussion Proceedings, IBERAMIA-SBIA 2000, Atibaia, Brazil, pp. 12–21 (2000)

Schiaffino, S., Amandi, A.: An Interface Agent Approach to Personalize Users' Interaction with Databases. Journal of Intelligent Information Systems 25(3), 251–273 (2005)

Schiaffino, S., Amandi, A.: Polite Personal Agents. IEEE Intelligent Systems 21(1), 12–19 (2006)

Shah, D., Lakshmanan, L.V.S., Ramamritham, K., Sudarshan, S.: Interestingness and Pruning of Mined Patterns. In: Proceedings of the 1999 ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD), Philadelphia (1999)

Shearin, S., Lieberman, H.: Intelligent Profiling by Example. In: Proceedings of the International Conference on Intelligent User Interfaces (IUI 2001), pp. 145–152 (2001)

Smyth, B., Cotter, P.: Surfing the Digital Wave: Generating Personalised TV Listings Using Collaborative, Case-Based Recommendation. In: Althoff, K.-D., Bergmann, R., Branting, L.K. (eds.) ICCBR 1999. LNCS (LNAI), vol. 1650, p. 561. Springer, Heidelberg (1999)

Sure, Y., Maedche, A., Staab, S.: Leveraging corporate skill knowledge - From ProPer to OntoProper. In: Proc. 3rd International Conf. on Practical Aspects of Knowledge Management, Basel, Switzerland (2000)

Villaverde, J., Godoy, D., Amandi, A.: Learning Styles Recognition in e-learning Environments using Feed-Forward Neural Networks. Journal of Computer Assisted Learning 22(3), 197–206 (2006)

Wiggins, J.S.: Personality and prediction: Principles of personality assessment. Krieger Publishing, Malabar (1988)

Xu, D., Wang, H., Su, K.: Intelligent Student Profiling with Fuzzy Models. In: Proceedings of the 35th Hawaii International Conference on System Sciences (2002)

Yang, J., Yang, W., Denecke, M., Waibel, A.: Smart sight: a tourist assistant system. In: Proceedings of Third International Symposium on Wearable Computers, p. 7378 (1999)

Yannibelli, V., Godoy, D., Amandi, A.: A Genetic Algorithm Approach to Recognize Students' Learning Styles. Interactive Learning Environments 14(1), 55–78 (2006)

Yasdi, R.: Learning User Model by Neural Networks. In: Proceedings of ICONIP '99, 6th International Conference on Neural Information Processing, pp. 48–53 (1999)

Yu, Z., Zhou, X., Hao, Y., Gu, J.: TV Program Recommendation for Multiple Viewers Based on user Profile Merging. User Modeling and User-Adapted Interaction 16(1), 63–82 (2006)

Zukerman, I., Albrecht, D.: Predictive Statistical Models for User Modeling. User Modeling and User-Adapted Interaction 11(1-2), 5–18 (2001)