Available online at www.sciencedirect.com
ScienceDirect
Computers & Education 51 (2008) 530–544
COMPUTERS & EDUCATION
www.elsevier.com/locate/compedu

Personalization in an interactive learning environment through a virtual character

E. Reategui [a,*], E. Boff [a], J.A. Campbell [b]

[a] Departament of Computer Science, Universidade de Caxias do Sul, 95070-560 Caxias do Sul, RS, Brazil
[b] Departament of Computer Science, University College London, Gower St., London WC1E 6BT, UK

Received 21 February 2007; received in revised form 29 May 2007; accepted 31 May 2007

Abstract

Traditional hypermedia applications present the same content and provide identical navigational support to all users. Adaptive Hypermedia Systems (AHS) make it possible to construct personalized presentations to each user, according to preferences and needs identified. We present in this paper an alternative approach to educational AHS where a virtual character personalizes the interaction with the user through the use of a particular recommender system. The character has natural language communication abilities; it can learn students' profiles and use this knowledge to recommend appropriate contents and activities. Through its interaction with the user, the character is able to collect and organize information about students in order to identify appropriate suggestions of contents. The recommender system employs a knowledge representation scheme that is easy to understand and to modify, enabling teachers/tutors to explore the types of recommendations being made and to appreciate why they are made. An experiment with computer science students was carried out in order to validate the approach proposed. The results of the experiment showed that the presentation of personalized links through a virtual character had a positive impact in the users' perception of the system as a learning tool. The combination of the virtual character with a recommender system proved to be a good alternative for the delivery of personalized contents without making constant changes in the main user interface. This approach provides mechanisms to guide users through paths of study followed by students with similar profiles, without violating the human–computer interaction principle of perceived stability.

Keywords: Virtual characters; Recommender systems; User modelling; Adaptive hypermedia

1. Introduction

The main principle of intelligent user interfaces is to enhance the flexibility, usability, and power of human–computer interaction for all users (Maybury, 2001). In order to do that, some of the problems addressed are: handling of information overflow; providing help for complex programs; taking over tasks from the user,

* Corresponding author. Tel.: +55 54 3218 2100.
E-mail address: ebreateg@ucs.br (E. Reategui).

0360-1315/$ - see front matter
doi:10.1016/j.compedu.2007.05.018

creating personalized systems (Ehlert, 2003). The work presented in this paper proposes answers for this last problem, i.e. taking into account differences between users and providing personalized methods of interaction. This is related to research in Adaptive Hypermedia Systems (AHS) where personalized presentations are constructed according to data collected through the interaction with users. Such systems have been built in many application areas, as in information systems, help systems and information retrieval systems, but the primary application of AHS has been in education (De Bra, 2000). Three functions are performed by an AHS (De Bra, Brusilovsky, & Houben, 1999): (1) While a user is browsing through a document, all interaction with the system is monitored. Based on the data collected, the system maintains a model of the user's knowledge about each domain concept; (2) The presentation of the document may be modified so as to make suggestions to the user about where to go next. Links can be added, changed, removed, sorted or annotated; (3) The AHS may conditionally show, hide, highlight or dim fragments of a page, ensuring that its contents include the appropriate information, at a suitable level of difficulty or detail.

Although the research on AHS has yielded promising results over the years, adaptive interfaces have received much criticism as adaptation and automatic assistance often contradict the principles of direct-manipulation and perceived stability (Tsandilas & Schraefel, 2004): How can the onscreen environment behave in a predictable and understandable way when changes are constantly made to the interface?

An answer to this problem is to shift from adaptive hypermedia to personalized virtual characters (André & Rist, 2002), i.e. anthropomorphic computerized beings whose main goal is to give the user assistance in operating a computer system and performing a given task. By using such characters as a communication channel, we may personalize the dialogues with the user and provide tailored assistance to each individual, without necessarily having to make major changes in the system interface. Nevertheless, virtual characters represent a big challenge to the research of adaptive hypermedia systems, adding complexity to the adaptation process and requiring the rethinking of the design of such interfaces.

At present, the use of virtual characters with different types of communication skills has spread in a wide range of applications, both in academic and commercial spheres (Pandzic, 2001). Within these applications, many examples can be found in the educational area. In Shaw and Johnson (1999), for instance, virtual teachers guide the students in online interactive activities. The interaction with animated and static virtual characters can also affect students' learning (Craig, Gholson, & Driscoll, 2002). In other research, it has been demonstrated that students considered the subject studied significantly less difficult and the presentation more entertaining in the presence of a virtual character (André, Rist, & Muller, 1999). In the same experiment most of the students stated that the assistants helped them pay attention to the most important details in the pages.

In this paper, we present an educational system which personalizes the interaction with the user through the delivery of content recommendations by a virtual character. The character communicates with users in natural language and recommends appropriate contents and activities to them, according to their interests and needs. A profile management system is used to collect and organize student information and discover behavioral patterns in the data collected. A natural language mechanism endows the virtual character with communication abilities. Finally, a recommender system complements the character's architecture, being employed to use the patterns identified to make recommendations of contents and activities.

The paper describes the use of the virtual character in an interactive educational environment, emphasizing its recommendation mechanism. Results obtained from an experiment with 53 computer science students enabled us to verify the effectiveness of our approach in identifying and presenting personalized links to the users. The results of the experiment are presented here, as well as conclusions and directions for future work.

2. Delivering recommendations through a virtual character

Most of the research in the area of recommender systems is related to finding optimal algorithms for the retrieval of personalized information (Breese, Heckerman, & Kadie, 1998; Geyer-Schulz & Hahsler, 2002; Sarwar, Karypis, Konstan, & Riedl, 2000). However, the simple use of a recommender system introduces human–computer interaction issues, as the composition of a personalized page involves the modification of its structure and the presentation of different contents to different users. Recommendations may be blended together with other items, or they may occupy a specific area of a page to emphasize the personalized selection of items. Depending on the domain and the specific application, a particular policy may be more appropriate. A third possibility is to use a virtual character to deliver the recommendations to the user. The main advantage of this approach is that employing the human image (or an anthropomorphic figure) may draw the user's attention and emphasize the recommendations without necessarily having to rely on an intrusive mechanism (De Angeli, Lynch, & Johnson, 2001).

Placing a virtual character in an interactive learning environment can have a strong positive effect on the students' view of their learning experience (Lester et al., 1997). This notion has been called the ''persona effect'', and is becoming an accepted precept among the computer–human interaction community. Adding intelligent and realistic behavior to these characters should improve their impact on the students' perception. For a banking application, it has been shown that virtual characters with personalization features can augment user involvement and reduce workload (Blom & Monk, 2001). To be precise, if the character is capable of identifying and treating the user according to his/her own features and needs, there is a strong probability that the user will be more engaged in the tasks proposed and will feel less strain in the work to be done.

Intelligent personalized agents to model the ability of customer-support assistants have also been proposed, being able to learn users' preferences and needs through machine learning techniques (Abbattista et al., 2002). An analogous approach is that of the Recommendation Battlers, where various virtual characters compete in the retrieval and presentation of the most appropriate information according to a search made by the user (Kitamura, 2004). The ideas presented in this paper are similar as far as the delivery of information through a virtual character is concerned. However, our approach is different in that we have employed a recommender system based on a particular learning and recommendation mechanism, explained later in Section 5.

An educational interactive environment in the domain of algorithms has been developed, with the main goal of making the courses more dynamic, and increasing the interest and participation of the students. The environment presents students with the regular contents of algorithms, proposes exercises, and provides a forum for discussion and a tool for the testing and running of algorithms (Fig. 1). Having been developed as a dynamic website, the system enables teachers and administrators to modify contents easily.

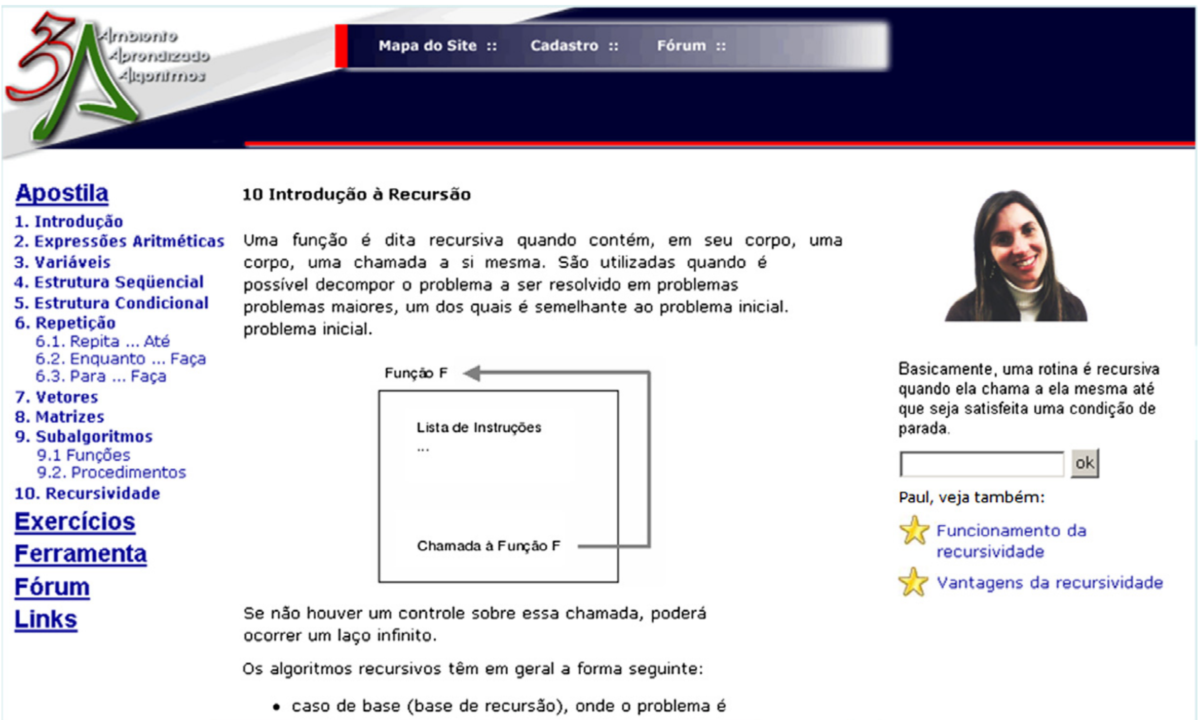

Fig. 1. Educational interactive environment in the domain of algorithms.

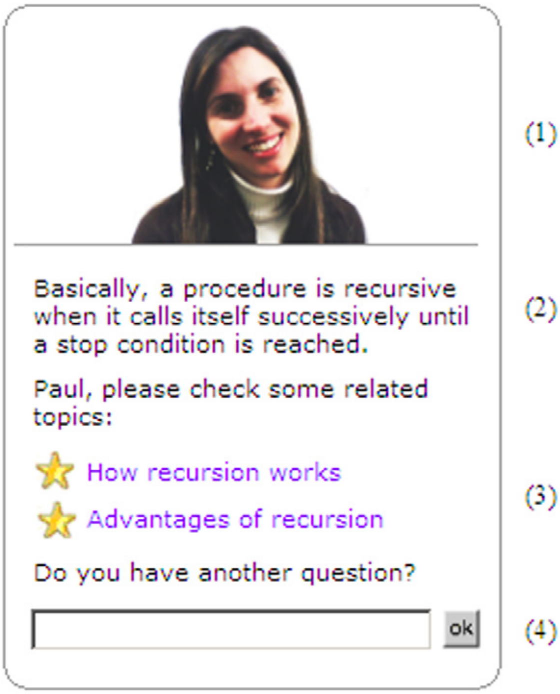

Fig. 2. Virtual character in an educational interactive environment.

A virtual character has been integrated to the system, being able to answer questions about algorithms in natural language, and to recommend personalized contents to each student. Fig. 2 shows the virtual character answering a question about recursion by displaying the information as on-screen text, and proposing other related material to a student named Paul.[1] The figure is divided into four sections, where we can see: (1) the image of the character, which changes according to the type of dialogue with the student; (2) the answer to the user's question through a natural language mechanism; (3) the recommendations about other topics of interest; (4) a textbox where the user may enter new questions.

The next section describes the architecture of the virtual character and details its main functionalities.

3. The virtual character architecture

Our virtual character has been integrated in an interactive educational environment to assist students in learning algorithms. Fig. 3 shows the architecture of the three main subsystems which compose the virtual character. The following sections explain each subsystem.

3.1. The profile management system

All the information that can be gathered about users is represented here in a hierarchical structure. Such structure lets one navigate through it in different levels of granularity, giving a clearer idea of the different information available for describing a user. This is analogous to the work of Middleton, Alani, Shadbolt, and Roure (2002) for instance, where the relations of document authorship and project membership are represented in an ontology and can be selected in order to identify communities based on publications and project work. Fig. 4 shows the ontology of a generic user model for an educational system in the area of programming.

[1] This website was originally created in Portuguese. It was translated into English in this example in order to illustrate the user interface for English-speaking readers.

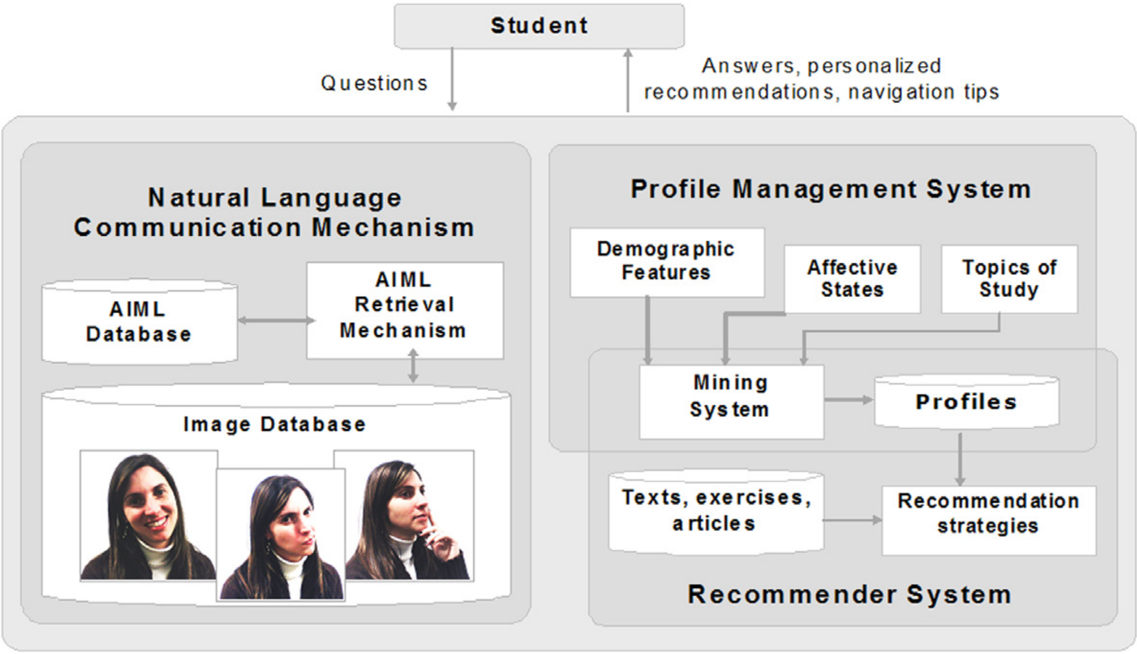

Fig. 3. Virtual character architecture.

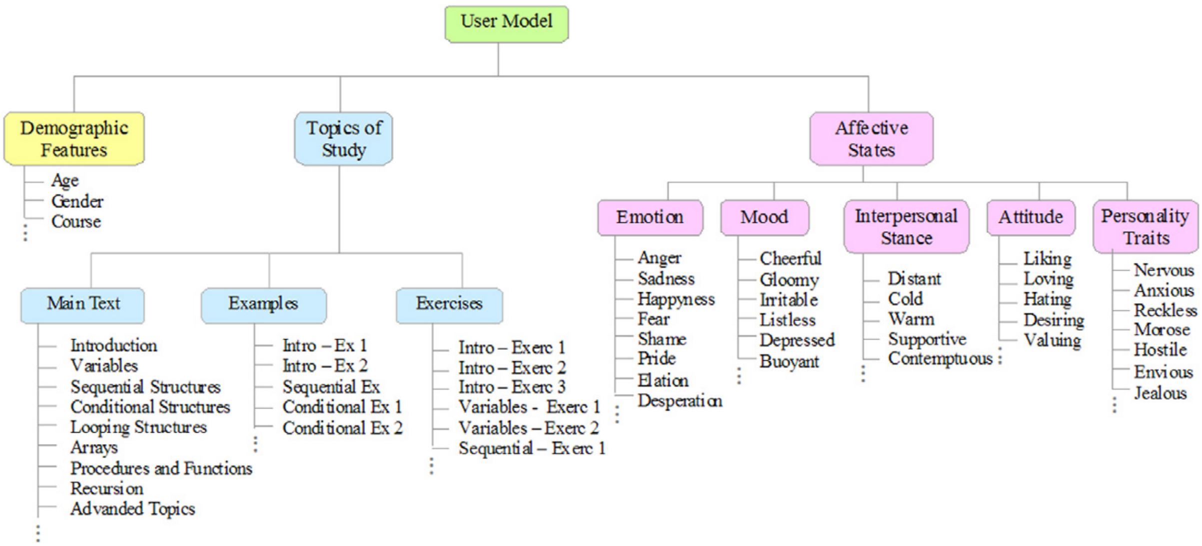

Fig. 4. Ontology to represent the user model.

Demographic features about users are read from a database and kept unchanged (unless more information becomes available). Examples of such features are: age, gender, course, hobbies, etc. *Topics of study* are the actual learning material available, such as *Texts*, *Examples* and *Exercises*. The access to these materials is monitored, letting the system distinguish the activities that each user has completed.

The third class represented in the upper part of the ontology is Affective States. It is intended to describe social-affective data which may interfere with the user's performance when interacting with the system. Affective states may be divided into five categories (Scherer, 2000):

- *Emotion* is the episode of evaluation of an external or internal event as being of major significance, relatively brief of synchronized responses for most organic systems. Examples are anger, sadness, happiness, fear, shame, pride, elation and desperation.
- *Mood* is a diffuse affective state that consists in a subjective feeling change, with low intensity, but long duration without apparent cause. Moods are considered to differ from emotions most strongly in not having an intentional object. Their causes are typically conceptual or evaluative (things are or are not going well). Some examples of moods are: cheerful, gloomy, irritable, listless, depressed, and buoyant.
- *Interpersonal stance* is an affective position in relation to the other person in a specific interaction. Distant, cold, warm, supportive and contemptuous are examples of interpersonal stances.
- *Attitudes* are relatively tolerant, affectively colored beliefs, preferences and predisposition in relation to objects or people. Examples of attitudes are liking, loving, hating, desiring and valuing.
- *Personality traits* are emotionally laden, stable personality dispositions and behavioral tendencies, typical of a person. For example: nervous, anxious, reckless, morose, hostile, envious and jealousy.

The cognitive approach to the modelling of emotions considers that different emotional states are attained according to evaluations based on world stimuli and the behavior of the individual (Ortony, Clore, & Collins, 1998). At present, there are four basic ways to recognize emotions through a computer system: voice; facial expressions; physiological signs (blood pressure, skin conductivity, etc.); and observable behavior. The latter corresponds to the observation of the user's interaction with the system, e.g. chosen options or typing speed.

Here, the system does not try to infer affective states, but the user reports explicitly how he/she feels at login time. This information is used to define the type of language and stimuli that our virtual character has to show in order to communicate better with the user. The next section explains its natural language communication mechanism.

3.2. The natural language communication mechanism

The knowledge base of the virtual character stores knowledge about algorithms, enabling the character to assist students mainly in theoretical questions. The Artificial Intelligence Markup Language (AIML) is used to represent the character's conversational knowledge (Wallace, 2003), employing a mechanism of stimulus-response. The stimuli (sentences and fragments which may be used to question the agent) are stored and used to search for pre-defined replies. The most important AIML tags are:

⟨aiml⟩ : indicates the beginning of a document.

⟨category⟩ : the simplest knowledge unit in AIML. Each category consists of an input question, an output answer and an optional context. The question, or stimulus, is called the *pattern*, while the answer is called the *template*.

⟨pattern⟩ : keeps a set of words which is searched for in sentences which the user may enter to communicate with the virtual character. The language that may be used to form the patterns includes words, spaces, and the wildcard symbols _ and *.

⟨template⟩ : when a given pattern is found in the input sentence, the corresponding template is returned and presented to the user. In its simplest form, a pattern is a word and the template consists of plain text. However, the tags may also force the conversion of the reply into a procedure which may activate other programs and recursively call the pattern matcher to insert the responses from other categories.

The optional context of a category enables the character to remember a previous statement. This feature, together with the possibility of launching particular programs when a certain pattern is found, makes the AIML communication mechanism very distinct from a simple retrieval of questions and answers from a database.

The user's affective state is also considered in order to choose the type of language the character uses to talk at a given moment. The affective state is entered as a pattern which has to be matched for the selection of a given sentence. For instance, the pattern *RECURSION* is modified into *RECURSION HAPPY* if the user is in a cheerful mood. Different sentences are retrieved for the two patterns, as depicted in the examples below.

⟨pattern⟩RECURSION⟨/pattern⟩
⟨template⟩ A procedure is recursive when it calls itself successfully until a stop condition is reached. Fractals illustrate well the use of recursion.
⟨/template⟩

⟨pattern⟩RECURSION HAPPY⟨/pattern⟩
⟨template⟩ Recursion is a cool topic! We call a procedure recursive when it calls itself successfully until a stop condition is reached. Fractals are a great example of the use of recursion!
⟨/template⟩

In addition to the existing AIML tags, new ones were created to manage the agents' emotional appearance. For instance, we created the tag ⟨humor⟩ to control the image changes reflecting different moods of the virtual character (happy, calm, aggressive, etc.). For each mood, several images are stored and selected at random, making the character more credible through a less repetitive behavior (Hayes-Roth & Doyle, 1998).

Therefore, when the user poses a question (stimulus), the character starts the AIML Retrieval Mechanism in order to build an appropriate reply using the information, patterns and templates from the AIML database. A picture of the character is picked from the Image Database to match the sentence retrieved according to the humor tag.

In addition to being able to answer questions in natural language, our character is also able to monitor the actions of each student and notice, for instance, that a particular topic is related to a given exercise. Such a behavior is achieved through the use of the *template* tag to launch the recommender system, which looks for appropriate activities and contents to each student. The recommender system component is detailed in next section.

3.3. The recommender system

Collaborative filtering is one of the most popular technologies in recommender systems (Herlocker, Konstan, & Riedl, 2000). The technique has been used successfully in several research projects, such as Tapestry (Goldberg, Nichols, Oki, & Terry, 1992), and GroupLens (Sarwar et al., 1998), as well as in commercial websites: e.g. Amazon.com Book Matcher, CDNow.com (Schafer, Konstan, & Reidl, 1999). The algorithm behind collaborative filtering (also known as social filtering) is based on the idea that the active user is more likely to prefer items that like-minded people prefer (Shardanand & Maes, 1995). To support this, similarity scores between the active user and every other user are calculated. Predictions are generated by selecting items rated by the users with the highest degrees of similarity.

Although collaborative filtering has been used effectively in a wide range of applications, it has a scalability problem: as for other techniques which rely on actual cases or records to arrive at a solution to a problem (e.g. case-based reasoning), the more users there are in the database, the longer it may take to find similar users and items to recommend (Linden, Smith, & York, 2003). Another drawback of collaborative filtering is that it is difficult to modify manually the way the system recommends items.

We have employed here a different approach which uses record-like structures, called item descriptors, to represent knowledge about how to make recommendations. This method is similar to content-based methods in which the system keeps descriptors of items, instead of storing information about users' preferences. However, rather than listing terms and keywords in the descriptors, user's features and item relationships are exploited.

3.3.1. The item descriptors

An item descriptor represents knowledge about when to recommend a particular item by listing the characteristics that users likely to be interested in the item should have. They are the main component used in the representation of the students' profiles. These characteristics can be classified as demographic (data describing

Game Programming | **Main Text**

Topics of Study		
Correlated Items	**Class**	**Confidence**
Introduction to Game Engines	Main Text	0.70
Game AI Principles	Main Text	0.66
C++ or Java?	Main Text	0.52

Demographic Features		
Correlated Features	**Class**	**Confidence**
Gender = "male"	Demographic	0.67
Age group = [18,25]	Demographic	0.42

Fig. 5. Item descriptor for the "Game Programming" text.

an individual, such as age, gender, occupation, address) or behavioral (data describing texts read, exercises made and preferences of an individual). It has been shown that both types of data are important when building a user profile (Buono, Costabile, Guida, & Piccino, 2001; Krulwich, 1997) and inferring users' preferences (Claypool, Brown, Le, & Waseda, 2001) . Let us examine an example of an item descriptor where behavioral and demographic features are considered to be relevant for the recommendation of a certain item (represented by descriptor *Game Programming* – Fig. 5).

The descriptor expresses the idea that users who are interested in the Topics of Study "Introduction to Game Engines", "Game AI Principles" and "C++ or Java?" will also be interested in the "Game Programming" text. In addition, it states that the majority of the users who have read the "Game Programming" text were male students in the age group [18–25] (Demographic Features). Each term's confidence factor is also displayed in the descriptor, representing the strength with which the term is correlated with the target item, in this case the "Game Programming" text. An explanation of how these factors are computed is presented in Section 3.3.2. A separate structure is used to keep the ontology for Affective States, Behavioral and Demographic Features (Fig. 6).

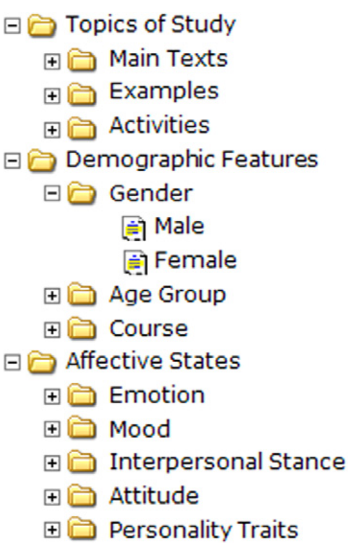

Fig. 6. Hierarchical information to represent the user model.

In this particular example, behavioral features have been represented in the class Topics of Study, which has been subdivided into three categories: Main Texts, Examples and Activities. When the students access these materials, their actions are monitored. The Demographic Features have also been subdivided into three categories, namely Gender, Age Group and Course (the programs in which the students are enrolled at the University), to represent characteristics that may influence the materials accessed by the user.

While behavioral features are learned over time, demographic data about users is read from a database and kept unchanged (unless more information about users becomes available). The hierarchy is useful to give a clearer idea of the existing classes and features, letting the user navigate through it at different levels of granularity.

The information related to the affective states of the users is not inferred by the system, but it is obtained from a graphical interface at login time, as depicted in Fig. 7.

At login time the users are asked about some of their *emotions, attitudes* and *mood*, which are relevant considering the exploratory and learning goal of the interactive activities. The *personality traits* which are more permanent are not requested here. Such information is collected when the users fill in their registration forms.

The affective states classified as *interpersonal stances* were not taken into account because they indicate how a person feels in relation to another person in a specific interaction. In our system, we consider only user-system interaction.

3.3.2. The learning process

Behavioral data and demographic features are represented in the model in the same fashion, through attribute-value pairs. Therefore, the learning algorithm treats them in the same way when determining the correlation among features and items.

The correlation factor confidence is used in order to determine how relevant a piece of information is to the recommendation of a given item. This is the same as computing the conditional probability $P(d_i|e)$, i.e. the probability that the item represented by descriptor d_i is rated positively by a user given evidence e. Such evidence may correspond to a user's feature or item rated by him/her. Therefore, the descriptors can be learned

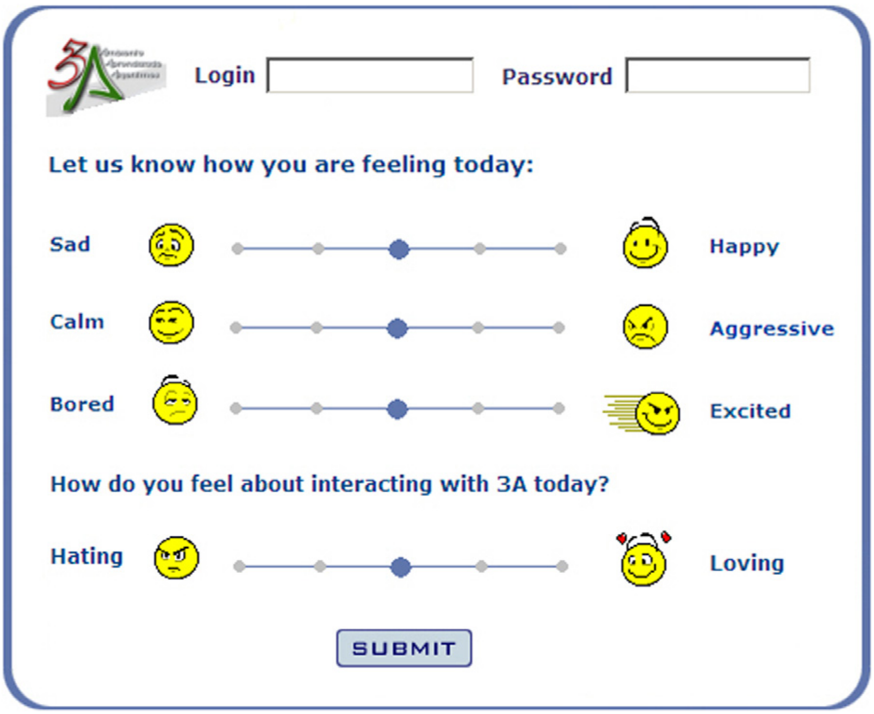

Fig. 7. Interface to collect the affective states of the user at login time. (This example has been translated into English in order to illustrate the user interface for English-speaking readers.)

through the analysis of actual users' records. For each item for which a recommendation strategy has to be built, a descriptor is created with the item defined as its target. Then, the confidence between the target and other existing demographic features and behavioral data is computed. This process continues until all descriptors have been created.

3.3.3. The recommendation process

The goal of the recommendation process is to find one or more items that match the user's preferences. Given a list of descriptors for n items $D = \{d_1, d_2, \ldots, d_n\}$, the recommendation process starts with the gathering of information about a given user to whom we want to make recommendations. All demographic features and behavioral data about the user are gathered and stored in a user record. Next, for each descriptor d_i, the list of terms $T = \{t_1, t_2, \ldots, t_k\}$ that match the user's demographic features and rated items is created. The system then computes a score for the descriptor d_i considering k matching terms. The score ranges from not similar (zero) to very similar (one), according to the formula below:

$$\text{(i)}\ Score(d_i) = 1 - \prod_{j=1}^{k}(1 - w_j \times P(d_i|t_j))$$

where $Score(d_i)$ is the final score of the descriptor d_j; $\prod$ denotes the product of an expression which considers each terms t_j present for descriptor d_i; w_j is the weight associated with term t_j; and $P(d_i|t_j)$ is the conditional probability of d_i given term t_j. The use of different weights for terms t_j enables the manual adjustment of the system to consider some items or features as more important than others in the computation of the descriptors' scores. In case all terms are considered equally important, the weights w_j can be set to 1, which is what we did in our experiment. In a practical example, consider a male user with student ID 00254, who is 26 years old and who has accessed some Topics of Study: "*C++* or *Java*" (*Text*); "*Introduction to Game Engines*" (*Text*); "2nd *Programming Exercise Lot*" (*Activity*). The information available about the student is organized in a user record, as shown in Fig. 8.

When matching the information about the user with the "*Game Programming*" descriptor presented in Fig. 1, three common terms are found:

- *Introduction to Game Engines* – Confidence 0.7.
- *C++ or Java*? – Confidence 0.52.
- Gender = "male" – Confidence 0.67.

Considering different weights for Behavioral and Demographic information (1.0 and 0.8, respectively), the $Score(d_i)$ representing how suitable is the recommendation of the "*Game Programming*" text for this particular user would be

Student ID: 00254

Topics of Study

Items	Class
Introduction to Game Engines	Main Text
C++ or Java?	Main Text
2nd Programming Exercise Lot	Activity

Demographic Features

Features	Class
Gender = "male"	Demographic
Age group = [26,33]	Demographic

Fig. 8. Information available about student ID 00254.

$$\text{(i) } Score(d_i) = 1 - \{[1 - (1 * 0.7)] * [1 - (1 * 0.52)] * [1 - (0.8 * 0.67)]\}$$
$$Score(d_i) = 1 - \{0.3 * 0.48 * 0.464\}$$
$$Score(d_i) = 0.933$$

This mechanism considers the complete history of topics recently accessed by the user as well as his/her demographic features in the recommendation process. Such an approach may be useful, for instance, to suggest texts and activities at the moment the user logs in. A slightly different approach is used, however, to look for one or more recommendations related to a single topic accessed at a given time. In this case, the system fills in the user record only with the user's demographic features and the specific item being accessed. The computation of the scores for the various item descriptors are then carried out as explained above.

Both approaches to recommendation are based on the assumption that any term matching the user's terms should increase the confidence that the descriptor holds the most appropriate recommendation. In a real-life example, suppose that we have a certain degree of confidence that a customer who buys a nail file will also want to buy nail polish. Knowing that this customer is a woman should increase the total confidence, subject to not exceeding the maximum value of 1.

This recommendation mechanism has been assessed previously in other domains (Reategui, Torres, & Campbell, 2004). When tested with the MovieLens database (http://movielens.umn.edu), storing anonymous ratings of 3900 movies assigned by 6040 users in the year 2000, the item descriptors performed better than the k-nearest-neighbor algorithm, regardless of the size of the neighborhoods chosen. The MSWeb database, available from Blake and Merz (1998), has also been used in validation experiments on the item descriptors. This database contains web data from 38,000 anonymous users who visited Microsoft's web site over a period of one week. Again, the item descriptors were more accurate than the *k*-nearest neighbor algorithm, and matched results obtained from other predictive algorithms applied to the same database and described in Breese et al. (1998).

4. Experimental results

An experiment was set up to evaluate certain aspects of the interactive learning environment related to the virtual character and its recommendation mechanism. A group of 53 undergraduate students was asked to use the system, all of them enrolled in the subject "Algorithms" in programs related to computer science at the University of Caxias do Sul, Brazil.

A preliminary version of the system was prepared with the main goal of collecting data to be used in the training of the recommender system. In this version the virtual character was present but no recommendations were made. A group of 13 students, around 25% of our sample population, was asked to study the topic of recursion with the interactive educational software, and to solve one out of ten exercises. In this group, there were 2 girls (15.4%) and 11 boys (84.6%). All of them had no or very little knowledge on the subject studied. After that, the students answered a questionnaire for the evaluation of the system. The data collected from the navigation of these students, totalling 482 transactions, was mined and used to fill out the item descriptors of the recommender server. The choice for a smaller number of participants in this first group had to do with the fact that we needed some initial data from which we could identify navigational patterns. The idea was to test how good/appropriate were these patterns in making recommendations for a larger group.

A second version of the system was prepared, this one capable of making recommendations according to the descriptors generated through the mining process of the initial experiment. The 40 remaining students were asked to do the same as the initial group, i.e. to study recursion using the interactive learning environment and to answer one of the exercises proposed. In this group, there were 7 girls (17.5%) and 11 boys (82.5%), giving the whole population a similar gender distribution. In this group, once again the students had very little or no knowledge on the subject studied.

The histogram below shows the awareness of this group of students regarding the recommendation of items (see Fig. 9).

Out of the 40 students, 8 did not notice any of the recommendations (20%, which could be considered an interface problem, i.e. the recommendations not being well emphasized). However, among the students who

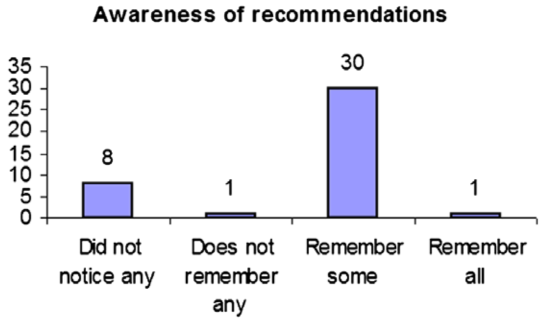

Fig. 9. Awareness of recommendations.

noticed the recommendations, almost all of them claimed to be able to remember some or all them. These results are convincing evidence that the recommender system was able to make suitable suggestions of contents to the students. To confirm this premise, the students were specifically asked in the questionnaire about the appropriateness of the recommendations. The answers are presented in the histogram below (see Fig. 10).

Considering only the students who noticed the recommendations, in the number of 32, all of them regarded the recommendations as appropriate (78.1%) or very appropriate (21.9%). These results enabled us to confirm the capacity of the recommender system to collect students' demographic and navigation data in order to find suitable content for recommendations.

We then compared the answers of the initial group (13 students who interacted with a character that did not make recommendations) with the second and larger group of 40 students who interacted with the character that recommended contents. There, we noticed the impact of the recommender system in the students' perception regarding the capacity of the character in assisting them. The histogram below shows the students' answers.

Fig. 11 shows that only 1 out of the 13 students (7.7%) of the first group thought the character to be helpful in carrying out the task proposed in the experiment. The character with whom the students interacted in this group did not have recommendation ability. For the second group, 25 out of the 40 students (62.5%) believed

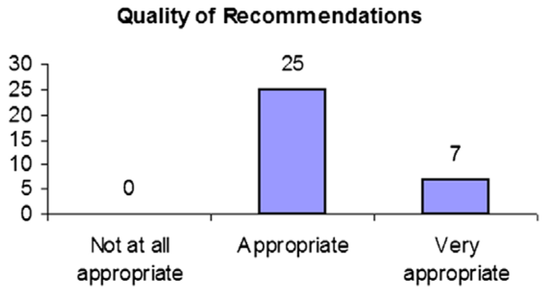

Fig. 10. Quality of recommendations.

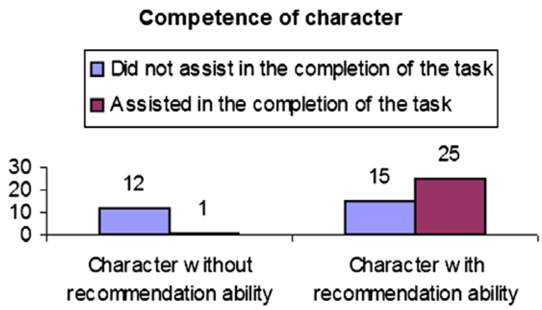

Fig. 11. Competence of character.

that the character was able to assist them in the completion of the task proposed. Note that the character available for this group had recommendation abilities. As the distribution of the population in the two groups was very similar, the difference in size of the two groups did not affect the analysis of the data and the interpretation of the results. The chi-square computed for this data set was 11.79 ($p \leqslant 0.001$), implying a significant distribution. This leads us to conclude that combining recommender systems with virtual interactive characters can improve the impact of the characters in the users' learning experiences.

5. Conclusion

We have presented in this paper an educational AHS where a virtual character personalizes the interaction with the user through the use of a recommender system. The main goal of the character is to identify and propose alternative navigation paths to each user through personalized suggestions of contents. In an educational context, such system is able to offer and tailor information proactively for individuals and communities, providing a mechanism to capture, to organize and to share knowledge.

Endowing the character with the ability to suggest personalized contents to each student improved the impact of the character in the perception of the users as to whether it was able to assist them in performing a certain task. This result brings up issues that can complement well-known character models, as in Hayes-Roth, Maldonado, and Moraes (2002), where the ability of a character to treat the user in a personalized way is not particularly emphasized.

The experiments carried out with an interactive educational environment allowed us to conclude that the presentation of personalized links through a virtual character had a positive impact on the users' perception of the system as a learning tool. By providing personalized contents through a virtual character, without making constant changes in the main user interface, the system is able to guide users through paths of study followed by other students with similar profiles, without violating the human–computer interaction principle of perceived stability.

Our decision to use the item descriptor approach instead of the more traditional collaborative filtering paradigm has been motivated by the fact that the knowledge learned in the descriptors is easy to understand and to modify. Such a feature may be important in interactive educational environments where the teachers/tutors may want to know the types of recommendations being made to students in each situation, and to appreciate why they are made.

Different types of information are considered in our student model, namely demographic, behavioral and social-affective data. While the first two types of data are used for recommendation purposes, the latter is employed to define the type of language and stimuli that our virtual character has to show in order to communicate better with the user. In some parallel work we are also investigating how *Affective States* and *Demographic Features* can be used in order to identify students who are more willing to cooperate with each other in an interactive environment supporting collaborative learning. There is similar work by Ahn, Brusilovsky, and Farzan (2005) who present a search system based on a social adaptive navigation mechanism exploiting the past usage history of users, motivated by those users' need for a social search capability.

The main contribution of the present paper has been to show how recommender systems can be combined with virtual characters in order to expand their abilities and improve the users' perception of the actual usefulness of the characters. Validating the recommender system based on item descriptors in an educational domain has been another important contribution.

We are currently working on other techniques to improve the interactive abilities of virtual characters. A recommender system based on social-affective information is also being worked on, in order to give the characters the ability to recommend student tutors to individuals showing difficulty in learning a given topic.

References

Abbattista, F., Degemmis, M., Licchelli, O., Lops, P., Semeraro, G., & Zambetta, F. (2002). Improving the usability of an e-commerce web site through personalization. In *Workshop on recommendation and personalization in Ecommerce. Second international conference on adaptive hypermedia and adaptive web based systems*, Malaga, Spain.

Ahn, J., Brusilovsky, P., & Farzan, R. (2005). Investigating users' needs and behavior for social search. In *Tenth international conference on user modelling. PIA 2005 – Workshop on new technologies for personalized information* Edinburgh, Scotland, UK. Accessed 24–25.07.2005.

André, E., & Rist, T. (2002). From adaptive hypertext to personalized web companions. *Communications of the ACM, 45*(5), May.

André, E., Rist, T., & Muller, J. (1999). Employing AI methods to control the behavior of animated interface agents. *Applied Artificial Intelligence, 13*(4-5), 415–448, May.

Blake, C. L., & Merz, C. J. (1998). *UCI repository of machine learning databases*, Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/~mlearn/MLRepository.html.

Blom, J., & Monk, A. (2001). One-to-one e-commerce who's the one? *Proceedings of CHI 2001 – Human factors in computing systems* (Vol. 31). Berlin: Sage, 31 March–5 April.

Breese, J., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the 14th conference on uncertainty in artificial intelligence*. Madison, WI: Morgan Kaufmann.

Buono, P., Costabile, M. F., Guida, S., & Piccino, A. (2001). Integrating user data and collaborative filtering in a web recommendation system. In S. Reich, M. M. Tzagarakis, & P. M. E. De Bra (Eds.), *Hypermedia: Openness, structural awareness, and adaptivity: International workshops OHS-7, SC-3, and AH-3* (pp. 14–18). Aarhus, Denmark, August.

Claypool, M., Brown, D., Le, P., & Waseda, M. (2001). Inferring user interest. *IEEE Internet Computing, 5*(6), 32–39.

Craig, S. D., Gholson, B., & Driscoll, D. M. (2002). Animated pedagogical agents in multimedia educational environments. *Journal of Educational Psychology, 94*(2), 428–434.

De Angeli, A., Lynch, P., & Johnson, G. (2001). Personifying the e-market: A framework for social agents. *Interact 2001 – Eighth IFIP TC.13 conference on human–computer interaction*, Tokyo, Japan, July 9–13.

De Bra, P. (2000). Pros and cons of adaptive hypermedia in Web-based education. *Journal on CyberPsychology and Behavior, 3*, 71–77.

De Bra, P., Brusilovsky, P., & Houben, G. J. (1999). Adaptive hypermedia: From systems to framework. *ACM Computing Surveys, 31*(4), December.

Ehlert, P. (2003). *Intelligent user interfaces: Introduction and survey. Research Report DKS03-01/ICE 01. Department of Information Technology and System*. The Netherlands: Delft University of Technology.

Geyer-Schulz, A., & Hahsler, M. (2002). Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory. In B. Masand, M. Spiliopoulou, J. Srivastava, & O. Zaïane (Eds.), *Proceedings WEBKDD'2002* (pp. 100–114). Edmonton, Canada, July.

Goldberg, D., Nichols, D., Oki, B. M., & Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Communications of the ACM, 35*(12).

Hayes-Roth, B., & Doyle, P. (1998). Animate characters. *Autonomous agents and multi-agent systems* (Vol. 1, pp. 195–230). Kluwer Academic Publishers.

Hayes-Roth, B., Maldonado, H., & Moraes, M. (2002). Designing for diversity: Multi-cultural characters for a multi-cultural world. In *Proceedings of IMAGINA 2002* (pp. 207–225). Monte Carlo, Monaco.

Herlocker, J., Konstan, J., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of ACM conference on computer supported cooperative work*, Philadelphia, Pennsylvania, USA.

Kitamura, Y. (2004). Web information integration using multiple character agents. In H. Prendinger & M. Ishizuka (Eds.), *Life-like characters: Tools, affective functions and applications*. Berlin: Springer.

Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. *Artificial Intelligence Magazine, 18*(2), 37–45.

Lester, J. C., Converse, S. A., Kahler, S. E., Barlow, S. T., Stone, B. A., & Bhogal, R. S. (1997). The persona effect: Affective impact of animated pedagogical agents. In *Proceedings of ACM CHI 97 – Human factors in computing systems*, Atlanta, GA, USA.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item to item collaborative filtering. *IEEE Internet Computing, 7*(1), 76–80.

Maybury, M. T. (2001). Intelligent user interfaces for all. In C. Stephanidis (Ed.), *User interfaces for all: Concepts, methods, and tools*. Maway, NJ, USA: Lawrence Erlbaum Asociates, Inc., Publishers.

Middleton, S. E., Alani, H., Shadbolt, N. R., & Roure, D. C. D. (2002). Exploiting synergy between ontologies and recommender systems. In *Proceedings of the 11th international world wide web conference – ACM WWW2002 – Sementic Web workshop*, Hawaii, USA.

Ortony, A., Clore, G. L., & Collins, A. (1998). The cognitive structure of emotions. Cambridge, UK: Cambridge University Press.

Pandzic, I. S. (2001). *Life on the Web. Software Focus Journal*. NY: Wiley, pp. 52–58.

Reategui, E. B., Torres, R., & Campbell, J. A. (2004). Using item descriptors in recommender systems. In *Nineteenth national conference on artificial intelligence (AAAI), proceedings of the workshop on semantic web personalization*, San Jose, USA.

Sarwar, B., Konstan, J., Borchers, A., Herlocker, J., Miller, B., & Riedl, J. (1998). Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. In *Proceedings of the conference on computer supported cooperative work*, New York, Seattle, Washington, USA.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommender algorithms for e-commerce. In *Proceedings of the second ACM conference on electronic commerce* (pp. 158–167).

Schafer, J. B., Konstan, J., & Reidl, J. (1999). Recommender systems in E-commerce. In *ACM conference on electronic commerce*, Denver, Colorado, USA.

Scherer, K. R. (2000). Psychological models of emotion. In J. Borod (Ed.), *The neuropsychology of emotion* (pp. 137–162). Oxford/New York: Oxford University Press.

Shardanand, U. & Maes, P. (1995). Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of human factors in computing systems – CHI* (pp. 210–217), Denver, Colorado, USA.

Shaw, E., & Johnson, W. L. (1999). Pedagogical agents on the Web. In *Proceedings of the third international conference on autonomous agents* (pp. 283–290). NY: ACM Press.

Tsandilas, T., & Schraefel, M. C. (2004). Usable adaptive hypermedia systems. *Hypermedia, 10*(1), 5–29, June.

Wallace, R. (2003). The elements of AIML style, ALICE A.I. Foundation.