PERSO: A system to build dynamically personalized courses in an e-learning environment

(ITHET 2003)

Henda CHORFI, Mohamed JEMNI
ESSTT, department d’Informatique
5, Av. Taha Hussein, B.P.56, Bab Mnara
henda.chorfi@esstt.rnu.tn
Mohamed.jemni@fst.rnu.tn

Esma AIMEUR
Departement d'informatique et de recherche operationnelle,
Pavillon Andre Aisenstadt, C.P. 6128, Montreal
H3C 3J7 (QC), Canada
aimeur@IRO.UMontreal.CA

Abstract - The aim of this work is to design an intelligent e-Learning Environment. We propose a system that initializes the student model with the affective and the cognitive information, which constitutes his background. When integrated in an e-learning platform, PERSO, offers to the student an individualized training program. The system is based on two approaches: Latent Semantic Analysis (LSA) and Case Based Reasoning (CBR). The first one is used to represent the curriculum and to analyze semantically the student responses. The second one is used to propose an individual training program to each student.

I. INTRODUCTION

With the spread of the New Technologies of Information and Communication (NTIC), we live a revolution in many domains particularly in education and with the development of a new mode of education: the e-learning [2, 3].

E-learning, the electronic delivery of information, communication, education and training, provides a new set of tools that can add value to all the traditional learning modes-classroom experiences, textbook study, CD-ROM and traditional Computer Based Training.

E-learning is a revolutionary way to empower a workforce with the skills and knowledge it needs to turn change to an advantage. It will not replace the classroom setting, but enhance it, taking advantage of new content and delivery technologies to enable learning.

However, in practice, and in the majority of cases, the change affects only the way how to deliver information (electronically) but not what to deliver. The most of web based training programs are just on-line documents. Trainers create electronic versions of traditional printed student manuals, articles and reference guides then diffuse them via dedicated systems called platforms.

Platforms are Web based systems that integrate services for virtual learning, teaching and informing including content creation, communication tools and user administration.

Most of e-learning platforms are focused on the transfer of knowledge, and this is where their shortcomings are evident. Knowledge sharing is very important, but it does not generate the behaviour changes needed to develop competent learners.

For these reasons, most of nowadays platforms are unable to detect the weaknesses of students, difficulties they meet and their needs. Furthermore, they are unable to react with them and to propose solutions to fix their problems.

In this context, the objective of this work is to grief intelligence in e-learning environments to make them more efficient. One of the main properties of an intelligent e-learning system, in our sense, is that it can built dynamically a suitable training program to each student corresponding to his preferences and his background.

In this paper, we start by providing an overview and a diagnosis of an e-learning experience we have recently done in the University of Tunis. Next, we give a presentation of the two main approaches used in the system PERSO: Latent Semantic Analysis (LSA) and Case Based Reasoning (CBR). Then we present the architecture of the system.
In order to explore the different aspects of this new mode of teaching and to analyze how it can be efficiently performed, the e-learning team of ESSTT (Ecole Supérieure des Sciences et Techniques de Tunis) achieved a pilot experience of e-learning (from November 2001 to February 2002). It developed two courses of MS-Word and MS-Excel and taught a group of 130 students (divided into 8 groups) of the first year of computer science bachelor. The courses were developed by teachers themselves with the help of two specialists in multimedia to treat images, audio and video sequences and to prepare flash animation and Java applets. The course has been taught by the Tunisian platform Waheeb. This later is a Web-based learning platform that provides a learning management system, and a range of content creation and publication tools. (For more details see [4])

At the end of the experience, we have performed two kinds of analysis: analysis of user’s appreciation (by mean of questionnaires distributed to both students and teachers) and analysis of statistical data delivered by the platform.

According to the questionnaires and the remarks of the professors, the results were promising and we have highlighted for the students, the following positive points: elimination of psychological barrier, discrete contacts with the teacher or colleagues (via e-mail or chat), possibility of feedback, respect of individual rate. In the same way, for the teachers: improvement of pedagogical methods, permanent availability of courses and flexibility of teaching schedule.

However two important questions emerged: First, why do we present a course in the same manner to the entire group since it is heterogeneous and all students do not have the same preferences? Secondly, why do we provide the same content and the same details to all students since all of them do not have the same background?

These questions emerged because with courses like MS-Word and MS-Excel, students may have an important, but not the same, initial knowledge. PERSO, the system we present in this paper addresses, directly, this problem and tries to match the student’s specific needs by recommending to every one an appropriate training program. This system is production-centred rather than learning-centred. It offers e-learning environments more efficiency and personalisation.

PERSO is based on two approaches: the Latent Semantic Analysis (LSA) and the Case Based Reasoning (CBR). These approaches are described in the two next paragraphs.

One of the main difficulties, when conceiving this system, is the choice of the appropriate technique to assess automatically free verbal statements. Unfortunately, progress in the area of computing the semantic content of texts has not been so successful. Two basic variants of semantic theories have been developed. One is based on some forms of logic. The other is represented by connections within semantic networks. In fact, the later can be simply converted into a logic-based representation [7]. Hence, the exploration requires the availability of a computational method that does a reasonable job of mimicking human comprehension. The recently developed Latent Semantic Analysis (LSA) technique meets this requirement.

In order to perform a semantic matching of pieces of text, LSA relies on large corpora of texts to build a semantic high-dimensional space containing all words and texts, by means of a statistical analysis [8].

Basically, the semantics of a word is determined from all of the contexts (namely paragraphs) in which that word occurs. For instance, the word bike occurs generally in the context of handlebars, pedal, ride, etc. Therefore, if a word like bicycle occurs in a similar context, the two words will be considered close to each other from a semantic point of view. Their corresponding vectors in the semantic space will be also close to each other.

LSA represents documents and their word contents in a large two dimensional matrix semantic space. Using a matrix algebra technique known as Singular Value Decomposition (SVD), new relationships between words and documents are uncovered, and existing relationships are modified to more accurately represent their true significance [7]. An interesting feature of this method is that the semantic information is derived only from the co-occurrence of words in a large corpus of texts. There is no need to code semantic knowledge by means of a semantic network or logic formulas.

LSA is fully automatic program, designed by Bellcore Labs from which we obtained the source code, written in C under Unix. We use LSA to represent the semantic space and to calculate the semantic similarity between the student answer and the right one (given by the professor).

The Case-Based Reasoning (CBR) is an area of exploration within Artificial Intelligence (AI). Studies of case-based reasoning begin with notions about how humans perceive the world and work out problems. A case-based reasoning system simulates these human methods in modeling real world problems. The premise of CBR is that once a problem has been solved, it often moves efficiently
to solve the next similar problem by starting from the old solution rather than by rerunning all the reasoning that was necessary in the first time. So the main idea of the CBR is to use the solution of a problem that has been solved earlier in order to solve a new one [10].

A case is the most basic element that represents an experienced situation. The CBR emphasizes the use of concrete instance over abstract operators. The techniques that make up the CBR are: case representation, indexing, retrieval techniques and adaptation. The CBR is a cyclic and integrated problem-solving paradigm, which uses the specific knowledge of previously experienced, concrete problem situations to find a similar old case, and reuse its solution in the new problem situation. In the system PERSO, a case represents the profile (both affective and cognitive profile parts) whereas, the solution represents the notions to be taught that the system recommends to the student.

V. SYSTEM ARCHITECTURE

PERSO is composed of five components: the curriculum, the student model, the analyser, the CBR system and the planer.

The following figure illustrates those components and their interactions.

![Fig. 1. PERSO architecture](image)

1) The analyzer

When the student connects to the system for the first time, he is asked to fill a questionnaire to determine his profile. This module generates that questionnaire and analyses the student responses.

Several types of questionnaires exists: the Exhaustive pre-tests (a large number of questions), the Adaptive pre-tests [3] (chooses the next question to ask the student by taking into account the answers to previous questions), etc.

The approach used in PERSO is categorization which is another way of avoiding to bury the student under a ton of questions [1, 6]. Each student has unique characteristics and behavior. However, it is often possible to observe patterns among students and to group students with similar features within categories, sometimes called stereotypes [9].

The categories we have observed are: beginner, intermediate and advanced.

In order to determine to which category the student belongs, the system performs one or two initial questions. These questions are about the entry points of the semantic network [1]. An entry point corresponds to the most advanced notion that the student masters in a given category.

To answer the question, the system gives the student two possibilities. If he does not know how to respond to the question he gives a pre-definite response “I do not know”. However, if he knows the response, he formulates it in form of free verbal statement.

When the student gives an answer, the system performs the analysis. Analysis consists of calculating the semantic closeness between the student answer and the right one (which is previously memorized by the system).

The system relies on three possibilities:
- if the semantic closeness $\in [–1; 0.5[$, the notion is covered poorly. It will be included in the course.
- if the semantic closeness $\in [0.5; 0.7[$, the notion is covered well. In this case, the system gives the choice to the student. The notion will be included in the course if the student desires it.
- if the semantic closeness $\in [0.7; 1]$ the notion is covered very well. It will not be included in the course.

The interval limits are variables and we can adjust them depending on the results given by the system (currently, we think that those values give the best results).

Example:

Give the different instructions to copy a block? (the system start with question corresponding to the entry point of the beginner)

```
Student response
  \{ I do not know \rightarrow Beginner
      \}
Analysis
  \{ Semantic closeness \rightarrow \}

\begin{array}{c}
\begin{cases}
\text{Beginner} & \rightarrow [–1; 0.5[ \\
\text{Second question} & \rightarrow [0.5; 0.7[ \\
\text{Second question} & \rightarrow [0.7; 1]
\end{cases}
\end{array}
```
Once the category is determined, the system is faced up to a new case. So, it performs new questions using similar previous cases, corresponding to the same category, stored in the Case Base.

2) The curriculum module

In this module, we represent:

a) The LSA semantic space: it symbolizes the context (it expresses human knowledge in a particular domain of interest). We developed a corpus of texts \( \text{(the course is a part of this corpus)} \) around the domain of computer science (domain of interest in order to teach, MS-Word). The text corpus is needed to build a LSA semantic space, which represents the knowledge that LSA has acquired from these texts. This semantic space can be reused for all courses dealing with the same domain. The semantic space is built by considering the number of occurrences of each word in each piece of text (basically paragraphs). For instance, with 300 paragraphs and a total of 2,000 words, we get a 300 x 2,000 matrix. Each word is then represented by a 300-dimensional vector and each paragraph by a 2,000-dimensional vector. The power of LSA is the reduction of these dimensions by a mathematical technique called SVD (Singular Value Decomposition). The reduced matrix permits all words and pieces of texts to be represented as N-dimensional vectors. (The best results are obtained with N comprised between 100 and 300).

b) The course: the knowledge representation used is a form of semantic networks, a graph where nodes are pieces of knowledge and edges represent relations between those nodes. The network of concepts forms the highest level of the knowledge base. The concept is made of five different kinds of elementary units: introduction, definition, example, exercise and recapitulation. The nodes in the network of concepts are linked by the following relations:

• « composition », enables to break up the teaching of a concept into the teaching of a succession of concepts. For instance, a teaching on “table manipulations” in MS-WORD can begin with “table insertion”, then “table modification”, next “table format” and finally “table conversion”.

• « equivalence » links concepts presenting equivalent notions. For instance, we link “text selection” and “table selection”.

• « prerequisite », enables to select what must be known to understand the concept. For instance, the concepts “table insertion” and “table modification” are prerequisite to the concept “table sorting”.

• « analogy », enables to understand a concept by analogy to another one. For instance, the notion “copy a block” is analogue to the notion “paste a block”

The following figure shows a part of the network of concepts.

![Fig. 2. A part of the Semantic Network of Concepts.](image)

In the network of concepts, we define three entry points according to each student category (beginner, intermediary and advanced). The semantic network is, in turn, divided into three sub-networks corresponding to each category.

The following figure shows the entry points to the semantic space.

![Fig. 3 Entry Points to the Semantic Network of Concepts](image)

---

\(^{1}\) This step is very important because the choice of texts affects the quality of LSA results.
3) The student model

The student model is composed of two sub-models: the cognitive model and the affective model.

a) The affective model: records the preferences in term of type of presentation. We propose four types of presentation: text/image, video, sound and simulation. We note that a notion is stored in the curriculum with different presentations when it is possible. To notify his most favorite kind of presentation, the student is requested to classify the different types of presentations from 1 to 4. For each notion to include in the course, the system tries to satisfy the first choice of the student. If this choice is not available in the database then, the system proposes the second choice and so on. The affective model is represented through a succession of entity-value couples to characterize the student preferences. Example: {(video, 1), (sound, 2), (simulation, 3), (text/image, 4)}

b) The cognitive model: As we said above, new students are not necessarily unfamiliar with all notions of the course. Our approach is to evaluate the learner background with a questionnaire called pre-test, which is given before he starts the course. The cognitive model represents the background of the learner concerning the different notions of the course. This part is implemented using an overlay model [1] that derives its structure directly from the structure of the course. The overlay model is made of semantic networks. Note that each node is a notion on which the student is questioned. A value is associated to each node that represents the semantic closeness between the student response and the right one.

4) The CBR system

In this module, we store the typical cases. A case is composed of the case number, the student identification, the student profile, the category and the different concepts to be included in the recommended course. The database contains three categories of students: beginner, intermediary and advanced. Students with similar profiles are grouped in the same category. Each category is initialized by three typical cases and updated, eventually, with new cases. When a new case arrives, the system try to find it’s similar cases to adapt a new solution for the current case. Eventually, this new case is stored in the case base (if it is considered to be relevant by the system). In this paper, we do not describe the CBR steps (indexation, retrieval, similarity and adaptation). (For details see [5]).

The figure below illustrates an example of a case in the beginner category.

![Fig. 4 A Case example (beginner category)](image)

5) The planer

This module proposes the solution to the student (an appropriate training program). The solution is a sequence of the different concepts to recommend to the student.

To generate a course, the system uses two processes: The first one extracts the notions and structures them in chapters. The second one deals with the type of presentation. For instance, the course that PERSO recommends to the student (corresponding to the case in the example above) contains all the chapters of the course. Each chapter contains all notions except for the first chapter, the notion “how to start Word”, does not appear.

VI. CONCLUSION

We presented in this paper PERSO, a system that can be integrated in an e-learning environment to give it more efficiency. It recommends to each student a personalised course.

PERSO is under development. It is based on a questionnaire in which responses are free text statements. A critical point for this kind of system is
a natural language processing mechanism that can robustly understand student input. LSA provides such a mechanism.

Another critical point when using questionnaires is to avoid burying the student under a ton of questions. With PERSO, we solve this problem by using categorisation.

Moreover, the advantage of using CBR techniques is twofold: on one hand, it allows us to minimise the number of questions to ask the student, on the other hand, it minimises the time for finding a new solution (personalised course).

Finally, the power of the system PERSO comes from the use of a combination of two techniques: CBR and LSA.

VII. REFERENCES


