PROTOTYPE IMPLEMENTATION OF AN INTELLIGENT E-LEARNING SYSTEM

Paul Dan Cristea, Rodica Tuduce, Ioana Alexandra Savescu, Cristian Alexandru Gorgorin,
Dan-Cristian Tomozei, Victor Radu Gradinescu and Calin Mihai Rangu
“Politehnica” Univeristy of Bucharest
Spl. Independentei 313, 060042 Bucharest, Romania

Abstract

Intelligent E-learning Systems may become more efficient and attractive to users by the specific support they can provide to both learners and tutors. The paper presents a prototype implementation of an intelligent e-learning system and some of the features it offers to the users.

Key Words

Intelligent e-Learning Systems, Learner’s Profile, Adaptive Environment

1. Introduction

In the past few years the use of information technology has increased substantially in the education domain. At the same time, expectations of students and teachers have increased as well. People from all over the world, with largely different cultural backgrounds and learning needs, can benefit from using Intelligent E-learning systems, that can adapt to their specific needs. Target public shifts gradually from full time students, to people already in mid-career, needing continuous education to improve their professional knowledge and skills. The requests can vary from executive or specialist up-dating to complementary or problem oriented study. Flexibility is an important issue when talking about e-learning systems. Developing easy-to-use tools for the tutors is another requirement that can determine the success or rejection of an e-learning system. Enhancing communication between all users is also important to help establish an co-operative environment that avoids isolating the user between the keyboard and the screen. The architecture of the system presented in this paper has been described elsewhere [ ], so that we focus on implementation aspects. The Learner’s Profile Eliciting Tool (LPET), based on the data resulting from the student tracking tool (STT), directs the Automatic Tutoring Tool (ATT) to adapt to the student objectives, preferences and current knowledge. The system recommends a path of study to the user in order to optimally approach his/her learning goals.

The paper presents the implementation of an Intelligent e-Learning System, discussing the roles of the users and the support provided to their fulfillment.

2. System implementation

The basic tools that have been used to develop the Intelligent E-learning System are a web server, a database server, scripting tools and a versioning server.

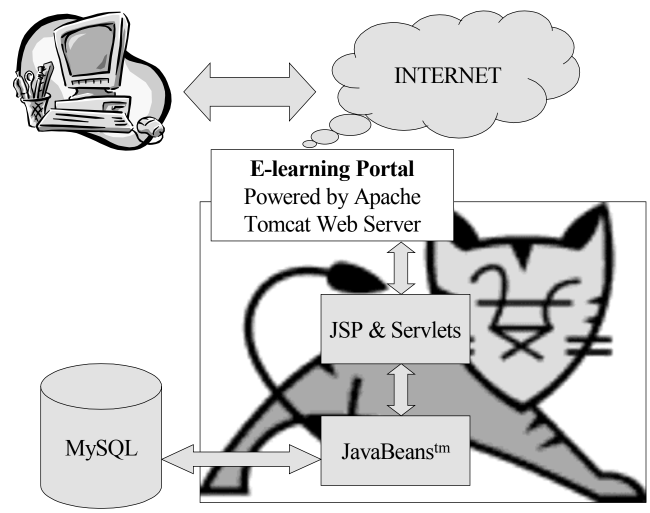

Figure 1 - General architecture

A general schematic of the current implementation is shown in Figure 1. User data access (some contained in the MySQL database) is made through interactive web pages implemented using JSP and Servlets. The architecture of the application is very simple for better performance results. The most relevant features of Java are used to accomplish security and database connectivity. The architecture is similar with MVC (Model View Controller) concepts: the interface is simple inside the JSP files (Java Server Pages); each JSP page is controlled by a JavaBean which performs the most important tasks of the page. Additional tools are used to handle files uploaded from the client on the server. Each JavaBean associated to a JSP file handles the security tasks and data manipulation.

Every JavaBean within a module extends a base class, one for each role (learner, tutor or administrator), thus knowing where to be redirected in unauthorized access cases (usually to the main page or the login page).

The system allows the change of the database by using a modular approach based on interfaces related to data sets. Only the module referring to the database is changed at the same time with it. Database access speed is increased by using a connection pool.

The database contains related tables that avoid redundant information. For example, a table keeps the available courses, the associated id-s, and the information about the owners. The table containing the learning items has a field that specifies the id of each course the learning item belongs to, without storing all the information about that course again.

3. Roles of users in an Intelligent e-Learning System

The user of an Intelligent E-learning System can have one of three major roles: administrator, tutor and learner.

The administrator may give rights (tutor or learner) to all users. He may see all tracking details resulting from the monitoring of learner's activity.

The tutor role involves responsibilities ranging from course content authoring and editing, to elaborating tests and quizzes for various learning items, and providing guidance to the learners. The tutor can structure the teaching material based on keywords attached to course items and questions. The tutor can adjust didactic requirements, like the number of points required to pass a learning item or number of questions in a test..

The learner is the central actor around which the system has to be organized. A learner is able to select courses for study and read different sections of the selected courses. He can also check his level of understanding for the learning items that he has studied by answering different questions.

3.1 Administrator role

The main responsibility of an administrator is to supervise the smooth functioning of the system. He has access to detailed information about the activity of every user. The report tool allows filtering and sorting data by several criteria (Figure 2). The implementation behind the filter form for reports is very general, so that it can be used in many other context. The filtering can be done taking into account all the fields of the report or only some of them.

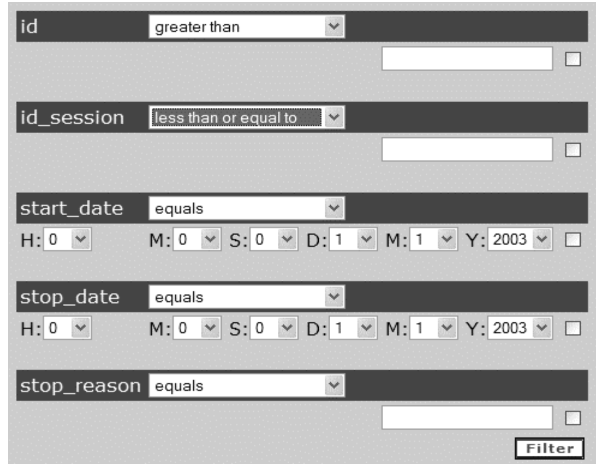

Figure 2 Filter form for reports

3.2 Tutor role

Each tutor has the right to modify the courses he has in charge. Once a course has been selected, the tutor's main authoring/editing window, shown in Figure 3, appears.

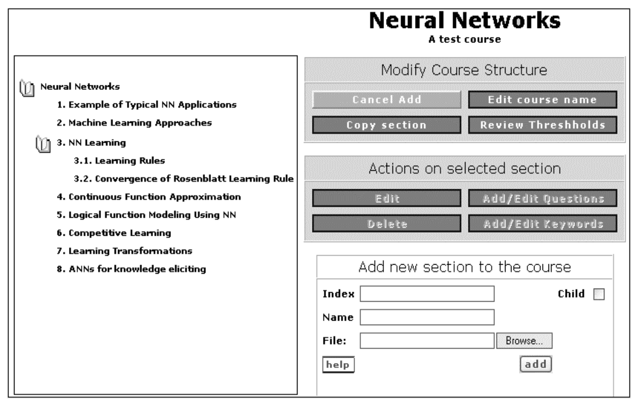

Figure 3 - Main authoring/editing window

The page is divided into 2 main frames. In the top-left corner the tutor can see the name of the course he is editing. Below, there is the table of content of that course, presented as a tree structure. The learning items that contain sub-items can be expanded or collapsed by clicking on the icon next to them. The right frame contains two sets of buttons: one for modifying the whole course structure, and the other for actions on a specific learning item (this set becomes active only when a learning item has been selected in the left frame).

The actions for modifying the course structure are "Edit course name", "Review Thresholds", "Copy section" and "Add section".

"Edit course name" allows the modification of the course name, or the description of the course.

"Review Thresholds" allows the tutor to set the minimum number of questions a test should contain for each learning item and the minimum number of points that a learner needs to obtain in order to pass each learning item. To make the job of the tutor easier and to help prevent awkward situations (like a quiz not offering enough points to pass the learning item), the system should suggest a possible interval for the threshold for a learning item (depending on the thresholds of all the available questions for it). Each question has a threshold T(Q); the sum of all the correct options PS(Q) and the sum of all wrong options NS(Q). The condition for T(Q) is:

$$PS(Q) + NS(Q) \le T(Q) \le PS(Q) \qquad (1)$$

The interval for the threshold of a whole learning item, T(LI) should be calculated as an average, considering the minimum number of questions to be generated for a quiz (LL → Lower limit of the interval, UL → Upper Limit of the interval).

$$LL = \frac{\sum_{i=1}^{NQ(LI)} [PS(Q_i) + NS(Q_i)]}{NQ(LI)} \cdot MNQ(LI) \qquad (2)$$

$$UL = \frac{\sum_{i=1}^{NQ(LI)} PS(Q_i)}{NQ(LI)} \cdot MNQ(LI) \qquad (3)$$

Selecting the threshold within this interval does not necessarily mean that every problems will be avoided (because of the random generation of the questions for each quiz), but gives the tutor an opinion about where it should be.

"Copy section" allows the copying of an existing section to another position in the course structure. The tutor should first select the item he wants to copy, then the item that will help determining the new position (either after it or as its child).

"Add section" allows the tutor to create a new learning item, by specifying a name, a position and a file containing the information. The file can itself contain the whole item (e.g. pdf format, doc format, html format without references, or even plain text), or it can be a zip archive, which should contain a file with the main information and other files containing images or other attachments. In the first case, the file will be uploaded on the server and its name stored into the database. In the second case, the zip archive will be processed on the server, all the files will be stored together in a folder and the name of the main file will be stored into the database. The database contains relative paths, for portability purposes.

The actions available for a selected learning item are: "Specify files", "Add/Edit Keywords", "Add/Edit Questions" and "Delete".

"Specify files" allows the tutor to change the file associated to an existing learning item. This is done in the same way as with the "Add section" action. It also allows the changing of the name of the learning item.

"Add/Edit keywords" opens a new page, allowing the tutor to add or remove keywords for the selected section. The new page displays in a frame the content of the learning item, making it easy for the tutor to select new keywords.

"Add/Edit Questions" opens a new page, allowing the tutor to add, edit or remove quiz questions for the selected learning item. The new page displays in a frame the content of the learning item, making it easy for the tutor to edit the questions. Each answer to a question can have keywords associated to it.

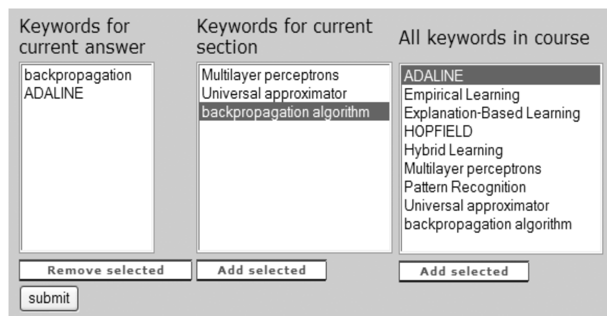

Figure 4 Editing keywords

The window displayed when editing keywords for an answer is shown in Figure 5. The tutor can select keywords from those associated with the current learning item or with the entire course.

The relation between the sets of keywords is:

$$K(A) \subseteq \bigcup_{LI \in C} K(LI) \subseteq K(C), \qquad (4)$$

where K(A) is the set of keywords for an answer, K(LI) is the set of keywords for a learning item and K(C) for the entire course. The set of keywords for a given course is obtained as follows:

$$K(C) = \bigcup_{LI \in C} K(LI) \qquad (5)$$

When a learner gives a wrong answer to a question, the keywords attached to that answer are checked. According to the relations between the sets of keywords (see equation 4) and the relation for K(C) (see equation 5):

$$\forall K \in K(A), \exists LI \in C \text{ so that } K \in K(LI), \quad (6)$$

where LI is a learning item and C is the current course. If the learning item is different from the current learning item, there are three possibilities:

- LI has been passed → the learner receives a warning advising him to review that LI
- LI has been failed → there is a penalty of points
- LI has not yet been tested → no action is taken

“Delete” deletes the selected learning item, after asking for confirmation. It will also delete all the files, questions and keywords associated with that section.

To help the tutor in operating the system, a help center should be available: to provide instructions, explain the necessary steps needed for accomplishing a certain action. This could be implemented either by “Help” buttons for every available action, or by an assistant able to answer questions from the tutor on how to work with the system.

For implementing a help assistant, a table is necessary to keep the possible answers and some keywords attached to each one of them. When the tutor types a question, all the words in the question are searched among the keywords and a list with possible answers is displayed. If no word in the question is found, the tutor is asked to reformulate the question.

3.3 Learner role

As shown in Figure 5, the learner’s reading screen is divided horizontally in two panels.

The larger right panel contains the current learning item displayed for reading and studying. The left frame contains an active table of content of the course, showing the standard path.

Below the frame containing the standard path, another panel is displayed with the recommended path, suggested by the Automatic Tutoring Tool on the basis of the Learner Profile. The recommended path can simply be a re-ordering or a selection of the standard table of content, but can also imply a more elaborate choice of one of several parallel learning items addressing the same subject in various approaches. The recommended path contains, besides the learning items that the learner has not yet passed, the learning items for which he has received a warning after having passed them.

For each learning item there is available a button: “Quiz”. When a learner presses this button, a quiz for the corresponding learning item is generated from the questions available in the database (there is a minimum number of questions for each quiz). Once the threshold for the learning item has been reached, it is displayed in a different color.

The learner is allowed to see the number of points accumulated for each learning item up to the present. He can reset the points for a certain learning item if he is not satisfied with his performance.

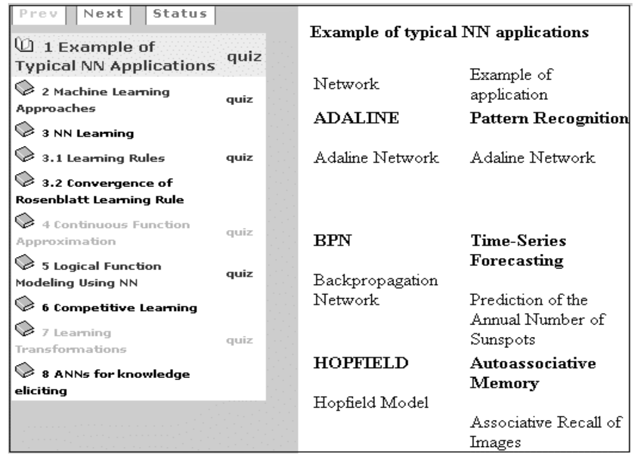

Figure 5 Learner's reading screen

Because negative points are assigned to wrong options, guessing is discouraged. Another fact that makes the results of the quizzes reflect the real level of understanding of a learner is that the quizzes are timed. This way the learner will not have the necessary time for looking through the course material during the test, being forced to answer the quiz with the knowledge gained so far.

4. LPET Architecture

The LPET is the dynamical component of the e-learning system. It represents the active element in the feedback loop used to calibrate the system’s response to student activity. A possible schematic representation of the system-integrated tool is shown in Figure 9.

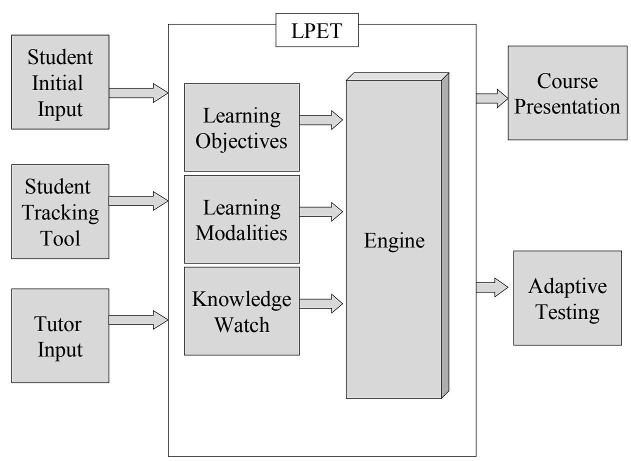

Figure 6 - Learner's Profile Eliciting Tool

The LPET gets input information from Student Initial Input, Student Tracking Tool and Tutor Input.

The Student Initial Input is, primarily, the user-submitted registration data. This information refers to personal user data and, more relevant to the LPET, to the course the student wants to attend and to the attendance's objective. The user's personal data is kept confidential.

The Student Tracking Tool keeps record of the student's evolution. It watches the student's activity, in terms of test and exercise results, as well as course reading progress and extra activities. The exercises are conceived in such a manner that they cover one or more sections of the course. Thus, a wrong choice in a quiz-question corresponding to a specific course section may show lack of knowledge in other course sections. As a consequence, the system should take the appropriate measures for the student to review the miscomprehended section. For every change in the teaching method, the LPET should also take into account the status of the student, how much of the course he/she completed and what parts are not yet passed. The extra activity involves discussions, especially between students and tutors and contribution to collaborative learning. For example a student can post new questions and exercises for the tutor to validate.

Tutors input is essential for the LPET, both as direct input about student's performance in learning and co-operating in the learners' group and as indirect input, when validating student submitted contributions to teaching items and quizzes. The parameterization of the course, by the choice of the points given for correct answers, the penalties for wrong options, and the thresholds to accept course parts as studied, is an important component of the whole teaching process by shaping the way the course is approached by the students. Moreover, the tutor's role consists in providing educational guidelines to be followed by students. Such guidelines can include information as which is the set of basic paths or sub-paths in accordance with some study programs. LPET will use this information to generate personalized paths based on the needs of the students. A student will find these as "recommended" paths, in addition to his/her other personalized paths. Another information which must be established by the tutor for training part of a curriculum is which parts must be considered as mandatory for a given class of students; this will in turn impose a constraint over personalized paths during their generation, forcing LPET to mediate between educational guidelines coming from teachers and courses customization coming from student needs.

All this input information leads to the development of a complete Lerner's Profile (LP), which holds The Learning Objectives, Learning Modalities and Knowledge Watch, and each of these components is computed by the engine of the LPET.

Learning Modalities (LM) refer to the desired learning style. For example, if the student prefers more practical-oriented or theoretical-oriented lessons about a given issue; this in order to select the desired level of detail and/or specific related materials (for example, more exercises) for each unit. Learning Objectives (LO) are closely linked to the LM, and involve the desired level of difficulty, e.g. basic, intermediate, or advanced. The LPET will use this to select the length and/or the level of detail of recommended paths. Knowledge Watch monitors the knowledge achieved by each student by analyzing quizzes results.

The LPET produces changes in the Course Presentation and Adaptive Testing. The way the course is presented to the student "adapts" by recommending him/her a more relevant order to study the given material. The quizzes may also be modified by the LPET by choosing questions from specific domains.

5. Communication facilities

From the tutor's point of view, the main drawback of e-learning systems is that they tend to isolate the student by hiding the teacher behind a machine. To solve this problem, each Intelligent E-learning System should provide a good communication tool. On one hand, the learner should have the possibility to write messages to the tutor of the course he is studying, either to ask questions or to make suggestions. On the other hand, the learner should be able to suggest possible quiz questions, which can actually end up in the quizzes, once the tutor validates them. This can allow the student to receive extra points, in case his suggestions are good. The tutor should also have the possibility to write messages to all the students studying each of his courses. He can either give answers to questions asked through messages, or redirect the learner to further reading on the subjects of interest.

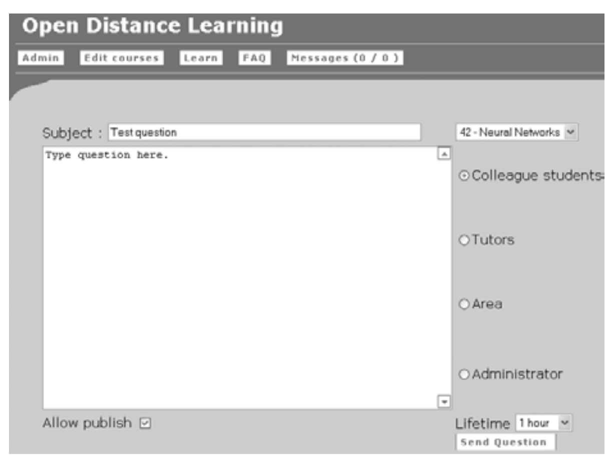

Figure 7 Message interface

Figure 10 presents the screen that allows writing and sending a message. The user can select the course which the question refers to, sending the message either to the

tutor of that course or to the learners that study that course. He can also write to the administrator of the system, to ask any questions specific to the system.
The report tool presented for the administrator role an also be used for sorting or filtering the messages for each user.

This module makes communication between students and tutors almost as easy as in the traditional, face-to-face learning system.

7. Conclusion

The paper presents a prototype system implementing the concept of Intelligent E-Learning Environments. The system has been developed in the framework of the Socrates Minerva project 87574-CP-1-2000-1-RO-MINERVA-ODL *Artificial Intelligence and Neural Network Tools for Innovative ODL* and is currently under testing and evaluation at the "Politehnica" University of Bucharest. Early results show that the concept of Intelligent E-learning System is positively received by both learners and tutors. The current implementation will allow gathering the feed-back needed for designing future versions of the system.

8. Acknowledgement

Part of this work has been supported by the Socrates Minerva project 87574-CP-1-2000-1-RO-MINERVA-ODL *Artificial Intelligence and Neural Network Tools for Innovative ODL* (http://www.dsp.pub.ro/).

References

[1] Vincenza Carchiolo, Alessandro Longheu, Michele Malgeri *Adaptive Formative Paths in a Web-based Learning Environment (*Educational Technology & Society 5(4) 2002, ISSN 1436-4522)

[2] Alexandra Cristea, P. Cristea, T. Okamoto, Opening the Blackbox of NN for Usage in the Modern Classroom, *Proc. ICCE'99,* Chiba, Japan, Nov. 1999, 787-788, *IOS Press, OHM, Ohmsha,* Amsterdam, Berlin, Oxford, Tokyo, Washington, DC, 1999.

[3] Alexandra Cristea, P. Cristea, T. Okamoto, Neural networks knowledge extraction., *Rev. Roum. Sci. Tech. – Electrotech. et Energ.*, *42*(4), 1997, 477- 491.

[4] P. Cristea, Adina Florea, Artificial Intelligence and Neural Network Based Tools for Cooperative Learning, *Intl. Workshop on Interactive Computer Aided Learning Tools and Applications*, Villach, 1999, 57-65.

[5] P. Cristea, Neural Networks as Tools for Knowledge Eliciting, *Proc. NEUREL 97 – 4th Seminar on Neural Network Applications in EE*, Belgrade, Yugoslavia, Univ. Belgrade, 1997, 2-9.

[6] P. Cristea, Alexandra Cristea, T. Okamoto, Knowledge Representation and Conversion, *Proc. ICCE'99 – 7th Intl. Conf. Computers in Education,* Chiba, Japan, November 4-7, 1999, 35 –43.

[7] P. Cristea, Alexandra Cristea, T. Okamoto, Knowledege Representation and Conversion, *Rev. Roum. Sci. Techn. Electrotechn. et Energ.*, *45*(2), 2000, 157-173.

[8] P. Cristea, Adaptation to Nonstationary Environments – Learning and Evolution, *Proc. IWALT 2000, Intl. Workshop on Advanced Learning Technologies*, Palmerston North, New Zealand, 2000, 265–267.

[9] P. Cristea, A. Arsene, B. Nitulescu, Evolutionary Intelligent Agents, *Proc. CEC 2000 - Congres on Evolutionary Computation*, La Jolla Marriott, San Diego, USA, July 16-19, 2000, 1320-1328.

[10] D. Dicheva, L. Aroyo, Alexandra Cristea, Collaborative Courseware Authoring Support, *WEBCATE 2002 – Computers and Advanced Technology in Education*, Cancun, Mexico, May 20-22, 2002, 52-57.

[11] J. Grudin, Computer-supported Cooperative work: History and Focus, *IEEE Computer*, *27*(5), 1994, 19-26.

[12] D.A. Norman, J.D. Spohrer, Learner-centered education, *Comm. of the ACM*, *39*(4), 1996, 24-27.

[13] M.J. Wooldrige, N.R. Jennings, Intelligent agents: Theory and practice, *The Knowledge Engineering Review*, *10*(2), 1995, 115-152.

[14] MySQL Database (2003). http://www.mysql.org.

[15] The Apache Jakarta Project (2003) http://jakarta.apache.org/tomcat/