TEST AUTHORING FOR INTELLIGENT E-LEARNING ENVIRONMENTS

Paul Dan Cristea and Rodica Tuduce
"Politehnica" University of Bucharest
Spl. Independentei 313, 060042 Bucharest, Romania, pcristea@dsp.pub.ro

Abstract

Intelligent e-learning environments (ILE) can significantly increase the attractivity of e-learning systems, as well as their teaching efficiency by adapting to each learner's profile (LP) and by providing multiple support to the tutor. The paper presents the authoring of tests for learner evaluation in an ILE, using the collaborative learning methodological approach.

Key Words
Intelligent e-Learning Environments, Authoring Tools

1. Introduction

Many of the currently available e-learning tools and environments are considered less appealing than the traditional face-to-face teaching methods by both students and tutors [1-3]. From the student's point of view, the main reproach is that rigid e-learning tools, which present the same static web pages to each user intending to acquire a certain item of knowledge, lack the "human understanding and intuition of the learner". Many current e-learning tools lack the ability to fit learner's specific objectives, interests, preferences, and even current disposition, usually included in the concept of "learner's profile", together with other features that define the student performance in learning a certain topic or in acquiring skills in a certain domain of activity. On the other hand, 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, while actually requiring more work form the tutor in preparing teaching material, assisting the students in learning, and assessing their advancement towards their established goals in terms of acquired knowledge and skills. A significant research and implementation effort [4-12] has been dedicated to develop Intelligent e-Learning Environments (ILE) able to optimally complement and/or replace traditional teaching, especially having in view life-long learning. Several European projects are currently focused on this topic. ILE components guide the trainee through the learning process, offer a platform for co-operative learning and knowledge discovery, and customize the presentation to learner's preferences, interests and needs. Artificial intelligence, connectionist and evolutionary tools are used to this end. On the other hand, adequate authoring tools are needed to help the tutors in preparing both teaching materials and tests for the students.

The paper briefly presents an ILE that implements the computer-supported collaborative work concept, stressing on the aspects referring to the authoring of the tests for student evaluation.

2. Intelligent e-Learning Environments

A web oriented intelligent e-learning environment (ILE) typically has a multiagent architecture as shown in Fig.1. Human and artificial agents collaborate to achieve the training tasks. Each learner is endowed with a Learner's Personal Assistant (LPA), permanently present in the system to assist the learner, to monitor his/her actions, and to assure coordination and communication with other agents in the system. The tutor has also a Tutor Personal Assistant (TPA) that performs similar actions, but with distinct aims. Specifically, TPA helps the tutor in keeping track of the class enrollment, presents essential data on learners' profiles and learners' progress in the training process, and allows updating of the course, exercises, tests and other teaching materials. Both the tutor and the student have full access to all teaching resources in the course and on Internet. Student actions, when logged-in to the system, are recorded by a Student Tracking Tool (STT) and passed to a Learner's Profile Eliciting Tool (LPET). The Learner's Profile (LP) is not conceived as a

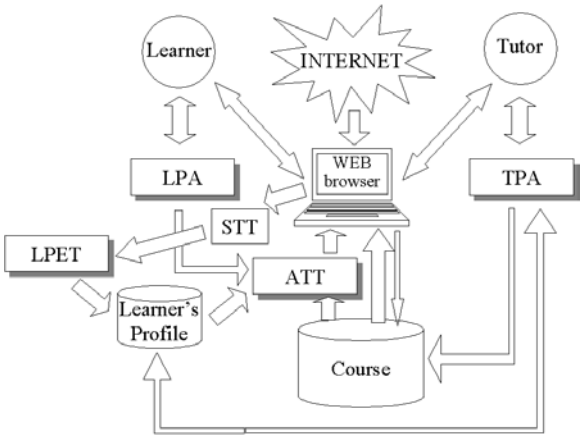

Fig. 1. Intelligent e-Learning Environment (ILE) typical architecture [5]

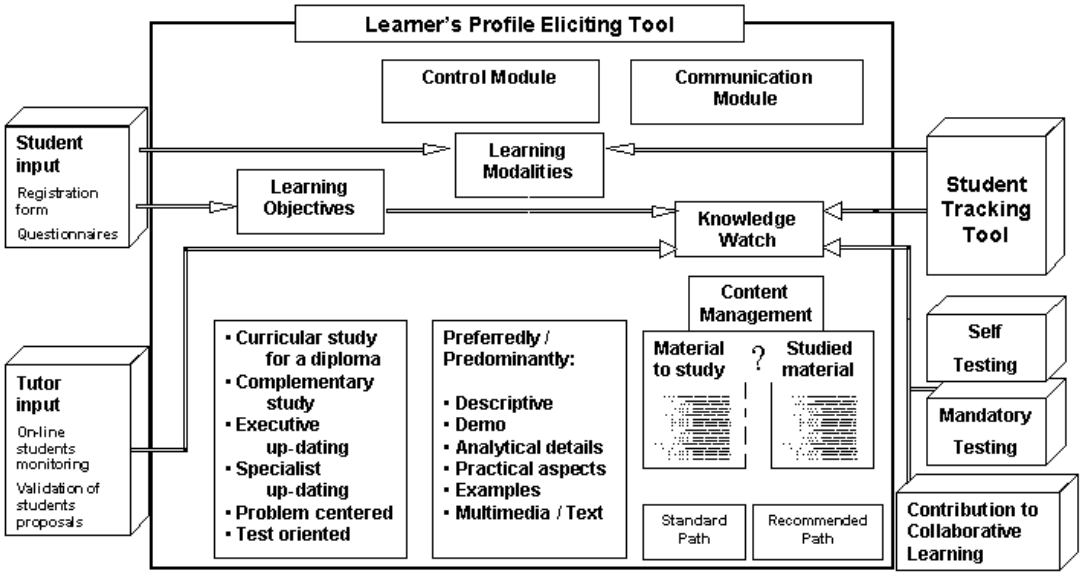

Fig. 2. Learner's Profile Model

psychological description of an individual intellectual aptitudes or performance, but as an operational data record of a student, used specifically for better presenting teaching material to that student, both for the same course and for other courses. Nevertheless, in accordance to privacy protection rules and legislation, the LP is treated as a private document, handled and used with due care. General conclusions, relevant to larger classes of students, without reference to individuals, are also used to improve the teaching. Fig. 2 presents a schematic diagram of the LP structure as extracted by the LPET showing its three basic components: Learning Objective (LO), Learning Modality (LM) and Knowledge Watch (KW).

The LO directly determines the basic requirements of the learning/teaching process. If the objective is a regular curricular study aiming at a diploma, the content of the material to be studied, the exercises to be undertaken, the tests to be passed must all satisfy some mandatory conditions for acceptance. On the contrary, if the learning is a freely taken complementary study, having the purpose of up-dating an individual's knowledge in a certain area and ending only with an attendance certificate, the content and the level of study is to a much larger extent the learner's choice. Even in such cases, there are significant differences between specific instances. For example, an executive up-dating, requires new basic facts and scientific/technical breakthroughs in a certain area to be presented in a digest-like manner. The purpose is to provide guidance in the field of interest, give fluency with the new terminology, clearly show new opportunities to allow strategic choices or decision taking, without going into technical details. On the contrary, a specialist up-grading must give a full working knowledge, even in a well delimited domain of interest, building on the previous knowledge of the learner. The LO also includes the case of problem centered training, when all materials, including tests, are structured around a certain task and are focused specifically at performing that task. The systematic use of keywords allows a flexible re-structuring of the teaching material, not only in accordance to the predefined hierarchical division and ordering chosen by the author when writing the course (*i.e.,* teaching items like course parts, chapters, sub-chapters and paragraphs), but also corresponding to the LO. The input used by LPET to build the LO comes primarily from the learner, starting with data in the registration form, completed by additional questionnaires, whenever necessary. The tutors and course administrators can complete and confirm the LO of each user.

The LM module is closely linked to the LO module as many aspects of the modalities of presenting the material are determined by the learning objectives. Still, there are aspects in the way the material should be preferably presented to a certain user that must be explicitly specified in the LM module: predominantly descriptive or, on the contrary, including analytical details and proofs, stressing on theoretical or practical aspects, following the construction required by the inner logic of the field (actually, the domain theory), or based mostly on demos and examples. The LM module also takes into account the technical aspects of the user (client) platform, i.e. screen resolution, usable bandwidth, processing power, to decide the relative weight of text, graphics and multimedia in the presentations in accordance to the available resources. To be really adaptive, an ILE requires a complex structure, with several parallel version of the same teaching item, to be selected in accordance with the LP. This approach can require a significant extra-effort in elaborating teaching materials, can involve several authors for various versions and might need institutional support, but brings the

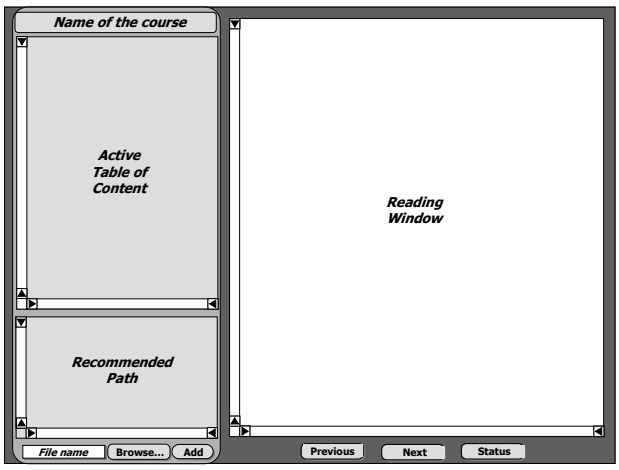

Fig. 3. Learner's reading screen

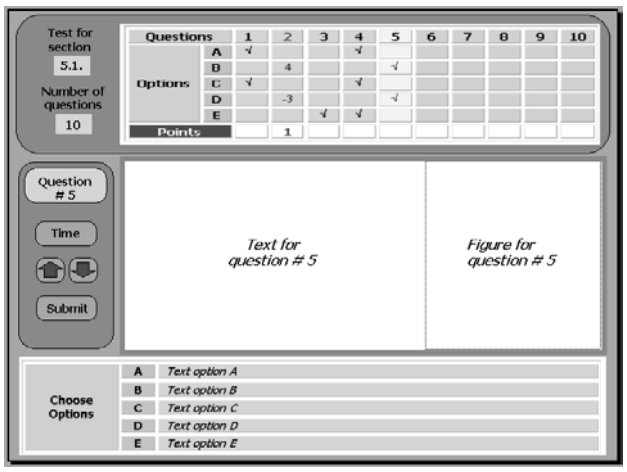

Fig. 4. Multiple choice test window

advantage of true flexibility and adaptation to the user interests. As a consequence, a course is no longer a flat juxtaposition of learning items, but a multilevel structure with many parallel branches, along which the ILE can recommend an optimal path for a user or for a class of users.

The KW provides the dynamic adaptability of an ILE for each user during the progress of the training. Basically, this section records the student's advance through the course material, as well as the level of mastering it, reflected in his/her performance in test passing and problem solving. As shown in Fig.3, the learner's reading screen is divided horizontally in two panels. Various control buttons and title lines are added to help the user navigate through the course. The larger right panel contains the current learning item displayed for reading and studying. The left panel is divided vertically into two sub-panels. The upper sub-panel contains an active table of content of the course, showing the standard path in a segment contiguous with the current open page or the last studied section. The already studied learning items are shown in distinct colors, depending on whether the points requested for the recognition of the learning have been partially accumulated, or the threshold for acceptance has been passed, or the tests have been failed. The lower left sub-panel contains the recommended path, suggested by the Automatic Tutoring Tool (ATT) on the basis of the LP. The data from the STT, the results of self testing and mandatory testing, as well as of the pro-active contribution of the student in supplying additional information to be added to the teaching material, additional tests or any other contributions in the process of collaborative learning are also taken into account in defining the LP and in choosing the recommended path. 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, depending on the priorities resulting from the LO and KW. The system is non-intrusive, as the student can follow either the recommended path or navigate along any path (s)he considers adequate, just by clicking the mouse button. Depending on the specific use of the ILE, several other intelligent agents and modules can be introduced in the system. Except for the tutorial material used in the learning by being told modality, all knowledge is organized around problem prototypes with associated specifications, generic solutions, and problem solutions. A dynamic structure of the teaching material can be achieved by using adequately chosen keywords. The interaction among the users of the system (learners and human tutor) is achieved either by direct communication through the network (forum and chat-like facilities) or by means of the Personal Assistant agents.

3. Learning appraisal

Learning appraisal is based on points obtained by the students in the assessment of their test results and their involvement in active learning. The ILE described in this paper uses a multiple choice questioning system for all the levels and learning items of a course. Figure 4 shows the aspect of the multiple choice test window, divided vertically in three panels. The upper panel comprises the information about the learning item (*LI*) to which the test refers to, the number of questions that must be answered and an active table that keeps track of the current state. For self-testing, both the *LI* and the number of questions can be specified by the user. In the case of mandatory tests, these parameters are usually imposed by the system/tutor. The questions can be seen either in the sequence of their numbers, relying on the default automatic sequential order of presentation used by the system, or can be opened randomly, by using the up/down arrows or by clicking the corresponding button in the active table. The background color of the columns in this table changes to make distinction between the already seen questions, the remaining ones, and the currently active question. The current question content is displayed in the middle panel and its available options for answers are shown in the lower panel. The options can be made by clicking the corresponding buttons either in the upper or

in the lower active tables. The corresponding field in the upper table is marked as checked. A previously answered question can be re-visited for changing the options before the submission, by pressing again the selection button in the upper panel After the submission by pressing the button "Submit" on the left of the screen, the "checked" marks are automatically replaced by the number of points obtained for each choice. The user can verify the result for the already answered questions or for part of them, before submitting the whole test, by pressing the button "Points" or the field in the Points row corresponding to the question. The points for the chosen questions will be shown in the table, like after the submission, but the options for these questions will be frozen.

Care is taken to stimulate students do more than just recognize textbook material when making their choices to respond to the tests. The proposed multiple choice questions require finer distinctions between correct and nearly-correct statements, based not only on plain recognition, but also on deductive and inductive inferences, on analysis and synthesis abilities. Higher-order thinking questions make the content of the questions challenging for the students and help them to understand thoroughly the course. Students are encouraged to formulate themselves new questions and are given credits when their contribution is included in the question data base (QDB). The questions are linked both with the hierarchically structured *LI* to which they refer, and with various transversal structures, dynamically defined by adequate sets of keywords corresponding to various explicit or implicit concepts, used especially in task oriented teaching. Not only each question and the corresponding correct choices, *i.e.*, the answers to the question, are tagged by one or several keywords, but wrong choices can have additional distinct keywords that signal a confusion or an error also relevant for another concept or learning item. A certain number of points is given or taken for each correct or false choice (option). The points are recorded for each question and each learning item and propagated upwards as shown in Fig. 5. There is a certain threshold which, when passed, the *LI* is recognized as accepted. The accounting is kept at all levels, for both the predetermined structure of the course (parts, chapters, sub-chapters and paragraphs) and for the dynamical conceptual structuring, in the case of the task oriented training. To make the system friendlier for the students and closer to a classic examination that does not require checking every item in a course, the passing of a higher level item is not conditioned by the passing of all lower level items it contains, but only by the sum of points exceeding the threshold for that level. On the other hand, the passing of a higher level item gives credit for all lower level items, below it in the course structure. Of course, this approach requests a carefully correlated selection of the number of points given for each answer, the number of the learning items at a certain level, and the acceptance threshold of their parent learning level. The acceptance of a part of the course is irreversible, so that only warnings and advices are issued if later errors signal

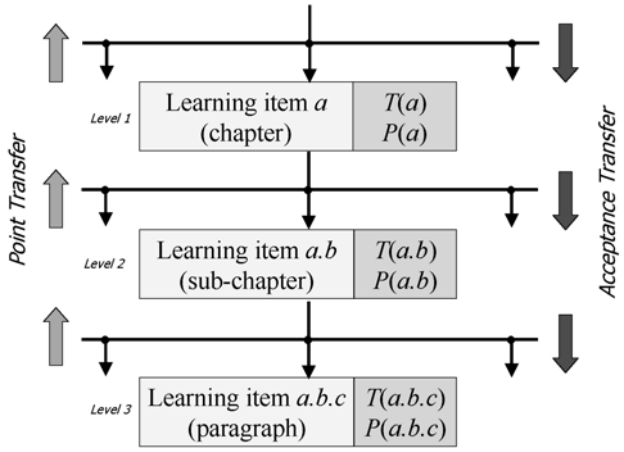

Fig. 5. Point and Acceptance transfer

possible weaknesses in an already assessed learning item. The tests are built by randomly choosing from the question data base (QDB) a specified number of questions referring to the given tag (*LI* address or keyword). When directly referring to a learning item the test can comprise questions attached either directly to that specific *LI*, or to any learning items placed below it in the course structure, *i.e.*, to its children $C(LI)$ and further descendants. The points obtained when making various choices C from the set of options $O(Q)$ pertinent to a certain question Q are recorded at the *LI* to which the question is attached and transferred upwards.

The sum of points for a question Q results by adding the points for all options selected at Q :

$$SP(Q) = \sum_{C \in S(Q)} P(C), \tag{1}$$

where $S(Q) \subseteq O(Q)$ is the set of options the student has selected at question Q. The correct choices (the answers to the question) are awarded positive points, while the wrong answers (the errors) bring negative points. Assigning negative points to the wrong choices contributes to discourage guessing. On the other hand, the sum $SP(Q)$ of points for a question Q is not directly the recognized result $P(Q)$ for that question. The amount of points $P(Q)$ acknowledged for question Q is given by:

$$P(Q) = \begin{cases} SP(Q), \text{ if } SP(Q) < 0, \\ 0, \text{ if } 0 \le SP(Q) < T(Q), \\ SP(Q), \text{ if } SP(Q) \ge T(Q), \end{cases} \tag{2}$$

where $T(Q)$ is the threshold required for the acceptance of the reply to Q. As a consequence, the positive points obtained for a question are taken into account only when exceeding a certain minimum requested amount, while negative points are always considered. Such a learning appraisal aims at a robust understanding and proper using of the tested knowledge, by discouraging superficial and fragmentary learning. On the other hand, each learning item is not evaluated independently, but in the context of the course, in relation with the related *LI*s. The sum of points $SP(LI)$ for a certain learning item *LI* consists not

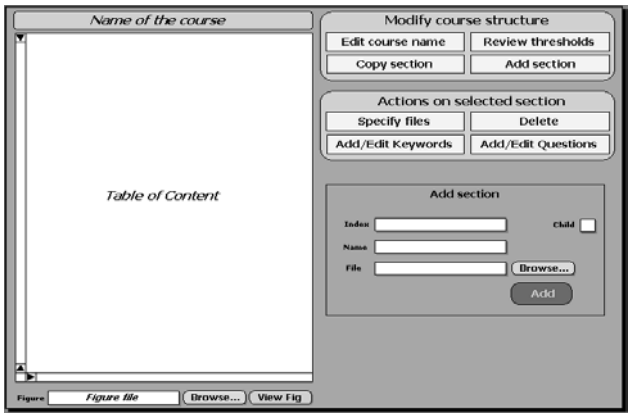

Fig. 6. Main authoring/editing window

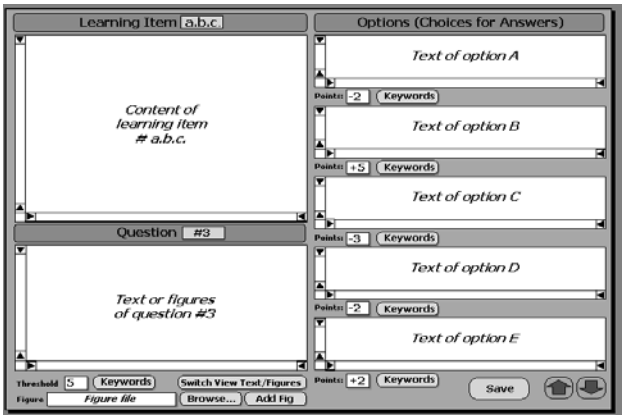

Fig. 7. Question authoring/editing window

only of the sum of the acknowledged points obtained for the questions *Q* referring directly to *LI*, but, also, of the sum of the acknowledged points transferred to *LI* from its children – the learning items *LI'* placed one level below it in the course structure:

$$SP(LI) = \sum_{Q \in LI} P(C) + \sum_{LI' \in C(LI)} P(LI'), \qquad (3)$$

with *C* (*LI*) – the children of *LI*. Equation (3) ensures that the points obtained for a certain learning item are transferred upwards, to all *LI*s that are its ascendants, without double counting.

The points *P*(*LI*) acknowledged for a learning item *LI* depend also on the passing of a certain acceptance threshold *T*(*LI*), but, in this case, there is an award *A*(*LI*) for the successful completion of the study of *LI* marked by the passing of *T*(*LI*):

$$P(LI) = \begin{cases} SP(LI), & \text{if } SP(LI) < T(LI), \\ SP(LI) + A(LI), & \text{if } SP(LI) \geq T(LI). \end{cases} \qquad (4)$$

At the same time, the status *S*(*LI*) of the learning item *LI* changes from 0 – *pending* to 1 – *studied*, when its threshold, or the threshold of its ascendant, have been passed:

$$S(LI) = \begin{cases} 0, & \text{if } \big(SP(LI) < T(LI)\big) \text{ and} \\ & \quad \big(S(LI') = 0, LI \in C(LI')\big), \\ 1, & \text{if } \big(SP(LI) \geq T(LI)\big) \text{ or} \\ & \quad (S(LI') = 1, LI \in C(LI')), \end{cases} \qquad (5)$$

with *C* (*LI'*) – the children of *LI'*.

Relation (3) shows the up-propagation of the points in the tree-like course structure, while relation (5) corresponds to the down-propagation of the acquired knowledge recognition along the same structure.

As mentioned, the acknowledged points can also comprise points for active co-operative learning activities. For instance, a student that submits a proposal for a new learning item – covering a complementary topics or bringing a new presentation for an existing topics, thus increasing the richness of the course parallel structure, receives additional points at the corresponding *LI*, if the tutor validates the proposal. Similarly, in the case the student submits a new question that is introduced in the QDB. The students are strongly encouraged to take active part in the learning process, with beneficial results for both his/her training and for the course diversity and dynamics. Figure 6 shows the main authoring/editing window, after clicking the "Specify files" button to add a new *LI*, while Fig. 7 shows the question authoring/editing window. The authoring tools for adding new *LI*s or new questions are accessible from both the learner's interface – the LPA, and the tutor interface – the TPA. The difference is that the proposal from the students are introduced in a buffer and submitted to the tutor. If the tutor validates the student proposal, the item is introduced in the adequate place in the course structure with a tag showing the student's name and the year, and the points are granted to the student.

4. Test authoring

Special tools are provided for question and test authoring. The authoring window gives direct access to both the teaching item for which the question is built and to the list of existing keywords. A special form receives all the details referring to a question. There are provisions to up-load figures as files attached to the questions. As mentioned, questions built by students can be seen and validated by the tutor, being automatically introduced in the QDB, while the corresponding points are automatically put on the student account.

As shown in Figure 7, the question authoring/editing window is divided horizontally into two panels. The upper part of the left panel shows the identifier of the *LI* to which the question will refer and displays its content. This is helpful in the construction of the question text. Scroll bars are provided to allow exploring the whole extent of the *LI*. If necessary, other related *LI*s can be displayed sequentially, by writing their identifiers in the corresponding text form field above the display window. A numerical identifier is automatically given to each new question, taking into account the already existing questions referring to the same *LI*. The text of the question can be introduced in the window below, by

direct typing, pasting, or specifying of a file containing it. One or several figures can be attached to the question body, by specifying the file(s) containing the pictures. The display can be switched form showing the text of the question to showing the figure(s). For this purpose, there is a button that cycles between the components of the question body. Below the editing window for authoring/editing a question, there is a text form field to introduce the threshold $T(Q)$ for the acceptance of the question Q (see equation 2). Near it, there is a button that opens a window for viewing, adding and editing the keywords that refer to the *LI* and Q. The right panel is divided in fields for introducing the text of each choice of reply to the question. The number of choices can vary, but a number of five options has been found to be satisfactory in most cases. The authoring/editing window for each option (choice) C is also foreseen with scroll bars. Below each window there is a text form field to introduce the number of points $P(C)$ for option C (see equation 3). The points for correct options (answers) are positive, while the points for wrong options (errors) are negative. Before allowing to submit a question, the system verifies that the fields for $T(Q)$ and for $P(C)$ of all existing choices C have been filled-in. It also verifies that the sum of the positive $P(C)$ exceeds or equals $T(Q)$, while the sum of all points is less then $T(Q)$. Messages of warning are issued if any of these conditions is not satisfied. For each choice, in addition of the keywords of the corresponding *LI*, which are inherited by the question Q and all its correct (positive) responses, it is possible to introduce specific keywords and links to other *LI*s for the erroneous (negative) responses. To this purpose, each choice field is provided with a button that opens a keyword window.

The window for test authoring is simpler, as it requests only introducing the identifier of the *LI* or the keyword for which to build the test and the number of questions. The questions corresponding to the specified parameters are chosen randomly from the QDB. The probability of the choice is decreased for the questions that have already been answered by the student. A warning is issued if there are not enough available questions for the requested test.

The question authoring/editing window allows reviewing all the questions for any *LI* and there is a tool allowing to hand pick the questions for a test by specifying their identifiers.

5. Conclusions

The paper presents an Intelligent e-learning environment (ILE) able to adapt to the learner's profile (LP) and which encourages active learning. A special attention is given to support the authoring of both learning items and tests, especially important in an ILE context to assess the student advance towards their learning objectives. Adequate authoring tools can make the tutor task easier, contributing to a better acceptance of e-learning systems, despite the extra work they require, and can stimulate the participation of students in the development of the system.

7. Acknowledgements

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

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

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

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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