Lamboudis, D., Economides, A. & Papastergiou, A.: **Adaptive exploration of assessment results under uncertainty**. *Proceedings 3rd IEEE International Conference on Advanced Learning Technologies, ICALT '03*, pp. 460-461, 2003.

Adaptive Exploration of Assessment Results under Uncertainty

Dimitris Lamboudis
University of Macedonia
156 Egnatia str.
Thessaloniki Greece
+30 231 891 768
dimlamb@uom.gr

Anastasios Economides
University of Macedonia
156 Egnatia str.
Thessaloniki Greece
+30 231 891 799
economid@uom.gr

ABSTRACT
In the context of Intelligent Learning Environments (ILE), adaptivity plays a key role. In order to achieve adaptive behavior an ILE should have a rich representation of the learning context, which is defined, among others, by the learner's characteristics, the type of the educational material, the advisory history, etc. Actually, the user model used by the system, and especially the representation and maintenance of user's knowledge can be considered as one of the critical factors that affect the system's effectiveness, in terms of it's capability to adapt to the individual learner's needs. In general, the evaluation of user knowledge derives from tests and tasks that the system proposes to the user to accomplish. This paper describes an approach that refines assessment results through user knowledge exploration, incorporating probabilities. We argue that the proposed approach leads to a better mapping of the assessment results to user knowledge in terms of its adaptivity to the response style of each individual learner.

Keywords
Intelligent Learning Environments, Adaptation, User Modeling, Pedagogical Agents, Assessment.

1. INTRODUCTION
Intelligent Learning Environments generally employ Artificial Intelligence techniques in order to adapt to the individual learner, and facilitate personalized learning [8],[9],[10]. ILEs are lying along a continuum, which runs from earlier "traditional" Computer Assisted Instruction systems (CAI), to very sophisticated systems that incorporate natural language dialogues, virtual reality, animated pedagogical agents, etc [1], [4], [5]. In the context of this paper we consider that these systems must consider a set of key decisions in their effort to support joint activity, including: *when* to engage learners with a service, *how* to best contribute to solving a problem, *when* to pass control back to users, and *when* to query users for additional information.

In order to reach such *situated* decisions, the system makes "guesses" about learners' needs, usually depending on the evidence obtained through the "keyhole" of the user interface, collaborative statistical data about the learner, explicitly asked information most commonly in the form of queries to the user in the beginning of a session, assessment evaluation, etc [3], [11]. The "intelligence" of a learning environment can be defined by its ability to make these decisions dynamically, at run- or user-time, based on an analysis of the learning context.

One of the main "ingredients" of the learning context is the learner and, from the system's point of view, the corresponding user model that the system maintains.

Student modeling remains at the core of ILE research. The student model stores information that is specific to each individual learner. At a minimum, such a model tracks how well a student is performing on the material being taught. Since the purpose of the student model is to provide data for the pedagogical module of the system, all of the information gathered should be able to be used by the tutor. Student models are generally considered to have three tasks: (i) They must gather data from, and about the learner; (ii) They must use the data to create a representation of the user's knowledge and learning process; (iii) The student model must account for the data by performing some type of diagnosis, both of the state of the student's knowledge and in terms of selecting optimal pedagogical strategies for presenting subsequent domain information to the student.

Especially we will focus on the system's capability to assess the current state of student's knowledge and the implied capability to do something "instructionally useful" based on the assessment.

One of the biggest challenges is to account for "noisy" data, the fact that students do not always respond consistently particularly when their knowledge is fragile.The learner's level of knowledge acquisition is evaluated by tests and/or tasks the user has to accomplish. That is, the responses of the user are mapped to its actual knowledge representation. Although different styles of scoring and mapping can be found, there is a common assumption made: a correct answer maps to knowledge, while a wrong answer maps to ignorance, faults, etc.

It can be argued, however, that this approach has two main shortcomings:

- in case of a correct answer, there is always a possibility that the user has answered by chance or at least he/she is uncertain about the answer chosen; it must be mentioned that the majority of the tests or the tasks in hand are, or could be seen, as multiple-choice questions; thus, with a question with five alternative choices, the possibility that a correct answer is the result of a guess is 20% - a possibility that cannot be ignored;

- in case of a wrong answer there is always a possibility that the user was mislead by factors irrelevant with his/her knowledge; for example, poorly designed questions, poor graphics in case the answer depended on them, etc.

Both cases lead to misconceptions about the actual user knowledge, which are difficult to be traced and revealed in the learning procedure to follow.

2. THE PROPOSED ALGORITHM

The proposed algorithm attempts to overcome some of the limitations that were mentioned in the previous section. The algorithm is engaged during a multiple-choice test, or in a task with discrete steps or sub tasks.

Instead of proceeding to the next question or task when the user provides an answer, the algorithm engages an exploration module allowing the user to have a second chance. This second chance is not provided unconditionally, since this would be equivalent to just adding more questions or tasks in the original design of the test, leading to a prolonged test that might not be ideal in all cases. Instead, when the user responds to a question, the algorithm decides to explore the answer's correspondence to actual knowledge by some probability P_e, and not to explore it by some probability $P_m = 1\text{-}P_e$. Thus, in the "worst case", the system will behave "conventionally", i.e. like in the existing systems. However, there is a possibility, which is partially defined by the designer, at least as far as the initial value of P_e is concerned, that the system will give the learner a second chance. Yet, if this possibility is heavily depending on the initial value of P_e, it would be just another ad hoc intervention of the designer, lacking any adaptive characteristics.

Instead, the probability of exploring user knowledge (i.e. the definition of P_e), is determined by the system, through the algorithm which checks if this exploration has any affects on the learning procedure, that is, if it reveals user knowledge that was previously hidden. In case it does, it reinforces the value of P_e, and in case it doesn't it decreases it. In the long run, this means that independently of the initial values of P_e and P_m the system will favour the option that actually helps the learner and the system to have a better representation of what the user actually knows. The corresponding notation and assumptions are as follows:

- A testing procedure that can be represented by a set of *n* ordered Questions or Tasks,
  $Q=\{Q_i, i=1... n\}$;
- An initial value of P_e^0 (the corresponding $P_m^0 = 1\text{-} P_e^0$); Where $P_e^0 = P$ (explore user knowledge/ given an answer);
- *Map* is a function that maps the answer of the student to his/her knowledge representation;
- *Explore* is procedure that is engaged to clarify user Knowledge, and
- *Update* is a function that updates the values of P_e^i and P_m^i;

The pseudo code of the algorithm is described below and its flow chart in Figure 1.

Pose Q_i to the Student

Given an Answer from the Student

By (P_m^{i-1}) Proceed to Map of this Answer to actual Knowledge or

By (P_e^{i-1}) Explore Students Knowledge

If (New Answer = Answer) then Map

Else Proceed to New Map

Update (P_e^i, P_m^i)

Proceed to Q_{i+1}

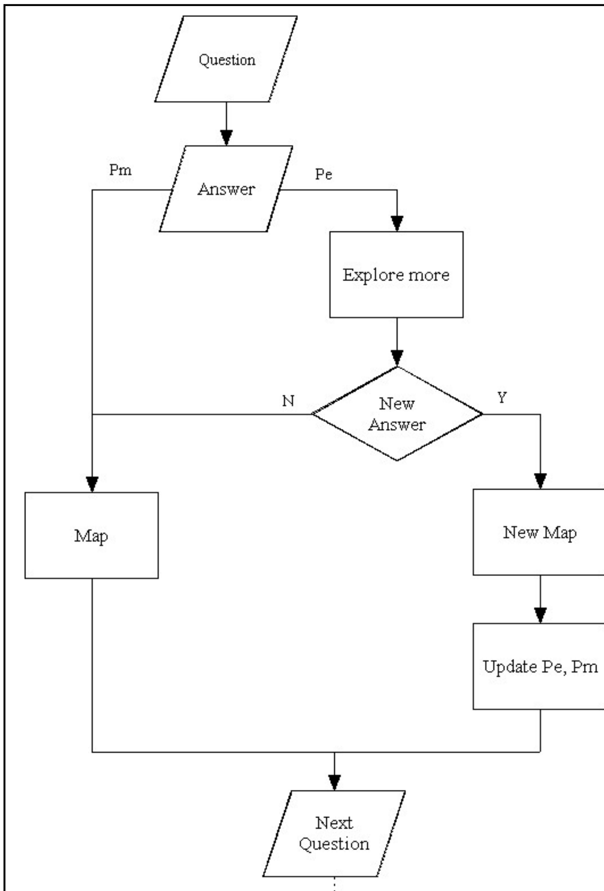

Figure 1. The algorithm's Flow-Chart

Up to this point the algorithm actually describes the intervention strategy that the system follows in order to clarify possible misconceptions about the user knowledge. The general idea is to increase the value of P_e^i if exploration results in a different answer from the original one, and to decrease it if the answer remains the same. Thus, in the questions to follow the system will favor the option that is meaningful for the user and the system itself, literally adapting its behavior to the individual learner. Of course, the updating function of probability values could be more sophisticated, and limitations should be incorporated to specify the upper and lower limits for P_e^i and P_m^i, depending on the particular implementation.

What is also depended on the specific integration of the algorithm and the designer's scopes are the exploration and mapping procedures.

For example, the exploration module in case of a correct answer, could ask for further details in the particular subject to check the validity of the original answer, or in case of a wrong answer to pose the question in a different style. As far as the mapping module is concerned, it could remain intact from the intervention of the exploration module (if there is one) and confront the new answers like being the original ones, or it could take into account that the answers are exploration results and apply a weighting strategy etc.

It must be noted that in this general form the algorithm can be integrated in systems that use the assessment procedure to trigger intervention from the systems side. For example, in case that an animated pedagogical agent is present monitoring the assessment procedure and acting correspondingly, the algorithm could be used to define if and how the agent will intervene. Moreover, this intervention will not only be useful to resolve misconceptions, but it will be done in a way that "hides" the behaviour pattern of agent (due to use of probabilities), thus enhancing its believability [7], [2], [6].

3. CONCLUSIONS

This paper proposes an algorithm that aims to enhance the systems ability to keep track of user's knowledge more reliably and more adaptively. Moreover, in case that it is used as part of the intervention strategy it could preserve the systems believability.

We have conducted some early experiments with students of our department to evaluate the algorithm. In particular we had our students run a simple multiple-choice test with and without the integration of the algorithm. This informal evaluation provided very positive results. Scoring was averagely 20% different, revealing lucky guesses but also not very clear questions.

Further work needs is currently under progress in the exploring and mapping modules in order to integrate a complete suite for assessment.

4. REFERENCES

[1] Andre E., Rist T., and Moeller J. (1999). Employing AI Methods to Control the Behaviour of Animated Interface Agents. *Applied Artificial Intelligence Journal*, *13 4-5*

[2] Bates, J. (1994). The Role of Emotion in Believable Agents. *Communications of the ACM, 37* (7), 122-125.

[3] Horvitz, E. (1999). Principles of Mixed-Initiative User Interfaces. *Proceedings of CHI '99, ACM SIGCHI Conference on Human Factors in Computing Systems, Pittsburgh, PA, May 1999.*

[4] Johnson W. L. (2000), *Pedagogical Agents. MIT Press*.

[5] Johnson W. L., Rickel J. W., Lester J. C. (2000). Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments. *International Journal of Artificial Intelligence in Education 2000,* 11, 47-78.Anderson, R.E. Social impacts of computing: Codes of professional ethics. Social Science Computing Review, 2 (Winter 1992), 453-469.

[6] Lamboudis D., Economides A., Managing Time Thresholds in Mixed Initiative Environments. *E-Learn 2002, World Conference on E-learning in Corporate Government, Healthcare and Higher Education, Montreal Oct. 15-19 2002.*

[7] Lester, J. C., Converse, S. A. Kahler, S. E. Barlow, S. T., Stone, B. A. and Bhogal, R. (1997a). The Persona Effect: Affective Impact of Animated Pedagogical Agents. *Proceedings of CHI '97 (Human Factors in Computing Systems,* pp. 359-366.

[8] Regian W., Seidel R., Schuler J., and Radtke P. Functional Area Analysis of Intelligent Computer-Assisted Instruction. *Training and Personell Systems Science and Technology Evaluation and Management Committee, USA*Conger., S., and Loch, K.D. (eds.). Ethics and computer use. Commun. ACM 38, 12 (entire issue).

[9] Sampson D., Karagiannidis, C., Kinshuk, (2002). Personalised Learning: Educational, Technological and Standardisation Perspective*. Interactive Educational Multimedia,* number 4 (April 2002), pp. 24-39.

[10] Shute V. and Psotka J., Intelligent Tutoring Systems: Past, Present and Future*. In Jonassen D. (ed.), Handbook of Research on Educational Communications and Technology. Scholastic Publications.*

[11] Zuckerman I., and Albreciit D. W. (2001). Predictive Statistical Models for User Modelling. *User Modelling and User Adapted Interaction,* 11.