

## ADAPTIVE STUDENT TEST EVALUATION INCORPORATING PROBABILITIES

by

**Athanasios Hatzigaidas, Anastasia Papastergiou, George Tryfon,  
Despoina Maritsa, Petros Galanidis**

**Abstract:** The aim in all CAT programs is to estimate the student's ability, performance and expertise level precisely and not to refresh or even better to improve the student's knowledge in a particular topic or subject. The priority of an ITS (Intelligent Tutoring System) or of a distance learning system or a hypermedia such system is the presentation, facilitation and support of learning. Consequently, it is of outmost importance the presence of an intelligent tutoring test, which could help the student to realize his ability level and expertise of topics of particular interest, through continuous tutoring testing. Such a tutor must incorporate student's profile, an intelligent agent, a large item bank for minimum item exposure and to use metadata for an easy and quick access to the knowledge pool. An Intelligent Tutoring Adaptive Tester capable of providing to the user the kind of assistance in order to maintain and even better to improve his/her ability level has been designed implemented and tested with encouraging results.

**Keywords:** Computer based adaptive testing, Intelligent tutoring system, Intelligent tutoring adaptive tester

### Introduction

The nature of computers as information processing tools, the role of computer technology in user-friendly interactive environments, and the possibility of designing instructional tools to meet individual needs of students, make computers potentially powerful tools for assessment (Shute V. et al; Morin, J. et al., 1997).

A computer adaptive test is a computer administers examination that adjusts to the ability level of the student taking the test as each question is presented (Wainer, et al; Weiss, et. al. 1984). Computerized testing is increasingly viewed as a practical alternative to paper and pencil testing. Some of the advantages of this kind of test are:

1. the assessment is targeted to the student's skill level
2. it provides a more accurate estimate of student's ability
3. the estimation is more precise using fewer questions

Computer based assessment applications used in science, such as Computer Adaptive Testing, Figural Response Testing, Sequential Computer Testing and Computer Simulations are currently used in large scale testing programs. Adaptive tests are comprised of items selected from a collection of items, known as item bank (Lord, F.M., 1980). The items are chosen to match the estimated ability level of the current test-taker student, with easier or harder items presented accordingly. This technique quickly converges into a sequence of items bracketing, and converging on,

the test taker's effective ability. Pilot testing new items can take place simultaneously with test administration.

## Design

The main reason of incorporating an intelligent tutorial adaptive tester and at the same time the main difference from a computerized adaptive tester is that a CAT program is used for TESTING while the ITAT is used for tutoring. In order to be able to carry out this task the system must include the following:

1. item bank
2. intelligent agent
3. student model and profile
4. metadata
5. knowledge bank

As the student advances in the learning process of a tutorial, the system via an exploration module will decide if the agent will intervene, and consequently the agent will decide through the second algorithm if he will pose a question from the previous tutorials and in which one of them (Figure 1). The student responses will be captured and analyzed by the system. The corresponding learning profile will be updated accordingly to become a personalized profile for the student concerned. When learners have achieved a rating of "4" for each tutorial in a particular tutorial group, they are assumed to have acquired a good enough understanding of that tutorial group and proceed to another.

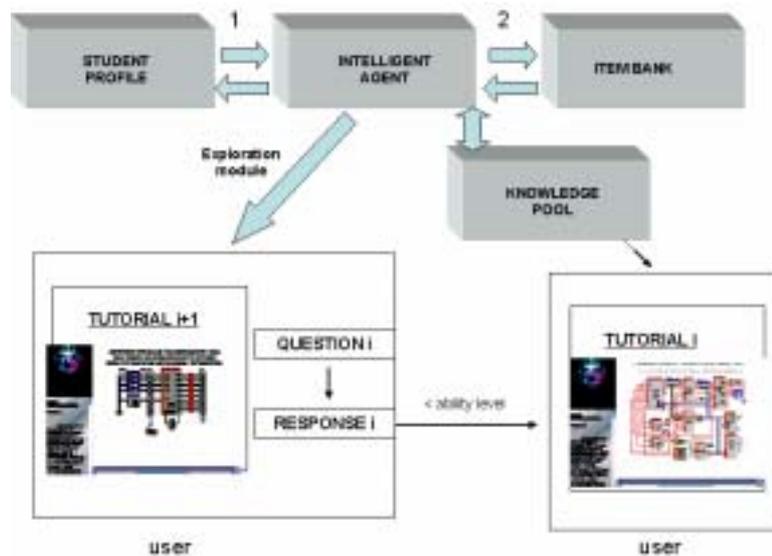


Figure 1: Schematic representation of ITAT operation

### **Item bank**

ITAT presuppose the availability of an item bank. An item bank is a structured collection of items. In developing an item bank we must determine size, dimensionality, response models, item removal and revision, adding items to the bank, maintaining scale consistency and bank calibration for adaptive tutoring testing.

The IRT model for the item bank that was used for the simulation studies reported on in this paper is the OPLM model (Hambleton, R.K. , 1991; Van der Linden, W.J. 1997).

In this model, the probability of correctly performing task  $i$ , also called the item response function, is given by:

$$p_i(\theta) = P(X_i = 1 | \theta) = \frac{\exp(a_i(\theta - \beta_i))}{1 + \exp(a_i(\theta - \beta_i))}$$

Here,  $a_i$  is the location parameter of the item. This parameter is associated with the difficulty of the item. This is the point on the ability scale where there is a 50% chance of correctly answering the item. Parameter  $\beta_i$  is the item's discrimination index. Estimates of the values of  $a_i$  and  $\beta_i$  for each item have been filed in the item bank.

### **Intelligent tutoring agent**

Tutoring agents are entities whose ultimate purpose is to communicate with the student in order to efficiently fulfill their respective tutoring function, as part of the pedagogical mission of the system. Educational applications, in general, are based on agents, even on web applications, where the focus is on distance learning (Johnson W. L., 2000; Lester, J. et al. 1997).

The examination Agent is part of the intelligent agent and plays a central role in the complete procedure. The examination Agent is responsible for updating the student results during the execution of an exercise (Morin, J. et al., 1997). This means that the Examination Agent will retrieve the new results of the student and subsequently add these results to the old results of the student. When the student finishes the exercise, the Examination Agent will derive a grade for the student and will write this to the student file.

### **Student Model**

The student model is a model of the user's characteristics and performance (Lester, J. et al. 1997; Zuckerman et. al. 2001).. This model contains data about all students that work with the program: all the information that is needed in order to

identify the students and to interpret the results of the students. The student model is defined in terms of elements that can be filled with data from an XML-file. Examples of data are student numbers, grades, and which exercises have been performed by the student. A list of exercises is kept for each student. This list contains all the exercises that the student has done in the past, and the results that are achieved for these exercises: the number of times that the student has tried the exercise and, when the exercise is finished successfully, the grade that the student received for the latest attempt. Also for each subtask, the number of tries is kept, and for each try if the student has passed the subtask successfully.

### **Metadata**

Metadata (data about data) will also be a key element of the future system architecture. Content must be associated with metadata to enable navigation, search filtering and adaptation. Metadata will assist in selecting the proper version of the content to be delivered, given the user preferences and device profiles. The process of adding metadata can, however, be very time consuming and resource demanding, especially for rich media content such as video, and research must be done to develop automatic or semi-automatic methods for adding metadata

### **Implementation**

The algorithm engages an exploration module to ask the agent to decide to act by some probability  $P_{ask}^i$  (Figure 2). Thus the probability of asking the agent to proceed to further exploration is determined by the system from the following proposed algorithm

$$P_{ask}^i = (10 - A_i) + k + L > 2 * k_{max} * c$$

Where

$k$  is a parameter representing the number of previous tutorials

$A_i$  is the learning ability of the student in a particular tutorial  $i$

$A$  is the students' average learning ability for all the previous lessons

$k_{max}$ : maximum tutorial number of the intelligent tutoring system

$L$ : represents the number of times the system requested questions to proceed to the agent. The system increases  $L$  for every time that the exploration module did not fulfill the request and takes the value of 0 when the exploration module will make a request to the agent

$c$ : random factor 0-1, generating through a random generator function

If the exploration module will decide to ask the agent to intervene, to explore more the student's knowledge in previous tutorial, the current tutorial attendance

stops, and the agent using all the necessary information obtained from the student model, the knowledge bank and the item pool and a new exploration module will determine if it will pose question to the student.

The agent explores the student profile, finds the related tutorials in which the student has level ability  $A_j < 7$  and via the new exploration module calculates the probabilities to pose a question for each previous tutorial.

The proposed algorithm for the new exploration module is as follows:

$$P_q^j = Q_j + R_j - T_j > c$$

where:

$c$  is a random factor 0-10, generating through a random generator function

$Q_j$  is the possibility to pose a question according to the level ability. Table I shows the predefined values which have been decided by the designer.

**Table I**

<b>A<sub>j</sub></b>	<b>Q<sub>j</sub></b>
A <sub>j</sub> > 7	0
A <sub>j</sub> = 7	3
A <sub>j</sub> = 6	5
A <sub>j</sub> = 5	7
A <sub>j</sub> = 4	9

$T_j$  represents the number of times that the system has re-asked question to the student from the specific tutorial

$R_j$  represents the previous failure of the student in reply from the specific tutorial

If the agent decides to pose a question from a tutorial, the student's response will be captured and analyzed by the agent, and the corresponding learning profile will be updated accordingly to become a personalized profile for the student concerned. If the students fail to answer the question the  $R_j$  parameter is updating. If the student fails again and  $R_j$  is equal to 2, he will be forced to re-attend the specific tutorial and at the end he will answer to another question.

In the next tutorial the exploration module will repeat the procedure and decide if the agent will intervene, and consequently the agent will decide through the second algorithm if he will pose a question from the previous tutorials and in which ones.

This way ITAT actually helps the learner improving his overall performance from previous tutorials and his learning ability through continuous revision.

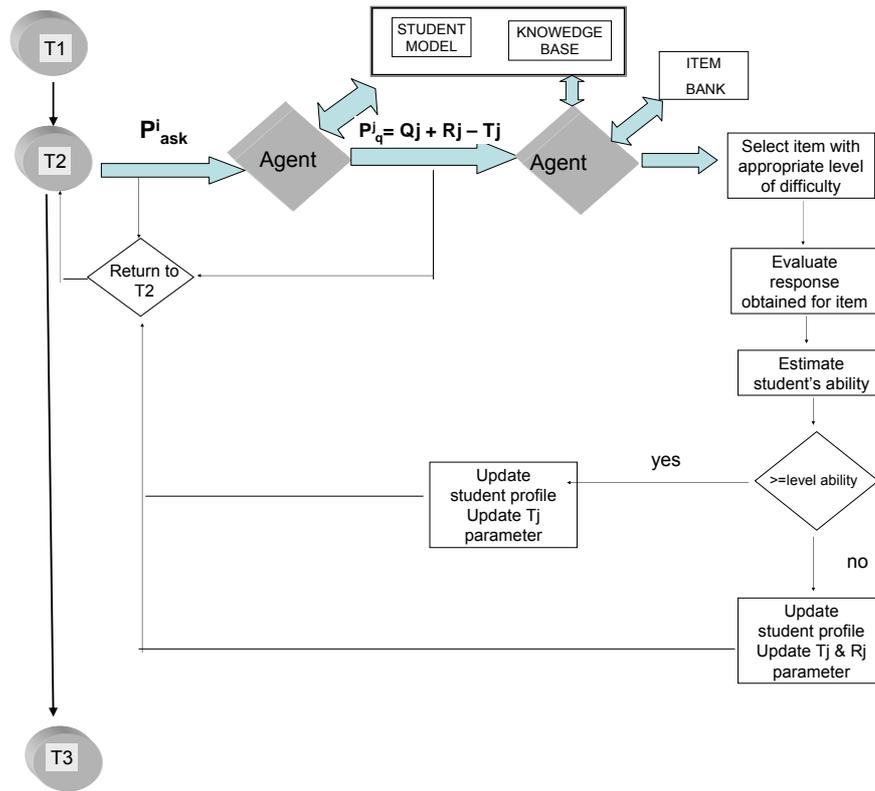


Figure 2: Flow chart of the ITAT

### Evaluation and Conclusion

The proposed ITAT has been implemented and embedded in an existing tutoring system (Figure 3) which has been developed and explored in the teaching and training area of micro controllers (Hatzigaidas, A. et al. 2003).

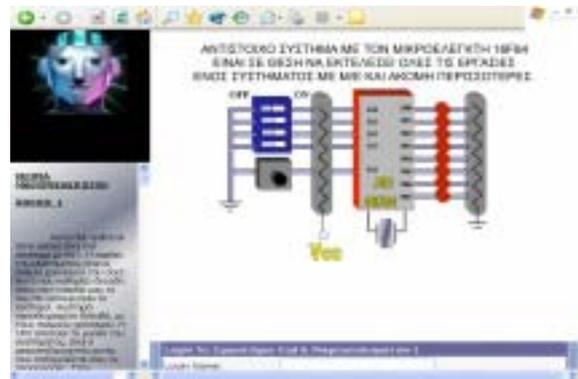
We have conducted some early experiments with students in the Department of Electronics to evaluate the proposed algorithms. This informal evaluation provided very encouraging results. Students were given the opportunity to interact with the system and explore its functionality.

At the end of the semester the students were interviewed in order to collect their comments, impressions and feelings about the system. They were asked to evaluate the system in a scale from 1 to 5, corresponding to 3 specific questions.

- Detection of student's educational need

- Adaptation to every student
- Functionality of the system

The majority of interviewed students had a positive response to the proposed system.



**Figure 3:** Overall of the tutoring system

Finally comparison between the level ability that each student obtained in the end of the tutorials attendance and the grade that student obtained in the classroom by attending the same tutorials and taking exams in a convenient way are similar, indicating that the viability of the proposed system is fair enough.

More work need to be done and is currently under progress in order to integrate the assessment results to user knowledge in terms of adaptivity to each individual learner.

## **Bibliography**

- Johnson W. L. (2000), Pedagogical Agents. MIT Press.
- Hambleton, R.K. (1991) Fundamentals of Item Response Theory. California: Sage Publications Inc.
- Hatzigaidas, A. Papastergiou, D. Lamboudis, G. Tryfon, D. Maritsa , P. Mentzelou (2003), Incorporating Educational Technology in Electronics Lab ED-MEDIA 2003 , World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, Hawaii, USA ,1, 381-384
- Lester, J. et al. (1997) Mixed Initiative Problem Solving with Animated Pedagogical Agents. AI-ED97 : Eighth World Conference on Artificial Intelligence in Education - Workshop V : Pedagogical Agents, 8.,1997. Proceedings.. Kobe: Japan.

- Lord, F.M. (1980). Applications of Item Response Theory to practical testing problems. New Jersey: Lawrence Erlbaum Associates (Publishers)
- Morin, J.F.; Lelouche,R(1997) Tutoring Knowledge Modelling as Pedagogical Agents in an ITS. AI-ED97 : Eighth World Conference on Artificial Intelligence in Education - Workshop V : Pedagogical Agents, 8.,1997. Proceedings... Kobe: Japan.
- Shute V. and Psothka J., Intelligent Tutoring Systems: Past, Present and Future. In Jonassen D. (ed.), Handbook of Research on Educational Communications and Technology. Scholastic Publications.
- Van der Linden, W.J. (1997). Handbook of Modern Item Response Theory. NewYork: Springer-Verleg
- Wainer, H., Dorans, N., Flaugher, R. Green, B., Mislevy, R., Steinberg, L. & Thissen, D. (1990) *Computerized adaptive testing: A primer*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Weiss, David J. and G. G. Kingsbury(1984,) Application of Computerized Adaptive Testing to Educational Problems, Journal of Educational Measurement 21, 4), 361-375,Perkins Periodicals J86JE
- Zuckerman I., and Albrecht D. W. (2001). Predictive Statistical Models for User Modelling. User Modelling and User Adapted Interaction, 11.

**Authors:**

**Athanasios Hatzigaidas**, sakis@el.teithe.gr,

**Anastasia Papastergiou**, natp@el.teithe.gr

**George Tryfon**, georgetryfon@freemail.gr

**Despoina Maritsa**, maritsa@hotmail.com

**Petros Galanidis**,

Technological Educational Institute of Thessaloniki, Greece