Incorporating Student Models in Adaptive Testing Systems

Avram Eskenasi, Tsvetelina Vladimirova and Julita Vassileva, *Sofia, Bulgaria*

SUMMARY

A method for using student models in adaptive testing is proposed. Two types of unsophisticated student model are used for achieving adaptivity: a model of the student's knowledge and a model of his or her individual characteristics. Three levels of adapting to the student are described: choosing the skills or concepts that will be tested, selecting the appropriate items for a fixed skill, and finding the appropriate form of the question of a fixed item. Different strategies for adaptive testing on these levels are described and the conditions in which any of them has advantages are shown. Obviously, further experimental work is needed to confirm practically the theoretical methods proposed. The next stage in our work will be to find methods for calculating the numerical estimate of the student's knowledge on the basis of his or her performance.

INTRODUCTION

After a large computer simulation study of student test-taking behaviour, Dosher and Bruno (1981) wrote: 'Results show test scores to be overstatements of subject-matter mastery, with large distortions at the lower achievement level'. In a later paper (Bruno, 1989) there is an even stronger statement.

How does one eliminate (or at least reduce) these distortions? Applying student models to adaptive objective testing may be one solution. Taking into account specific student characteristics one can try, as a first step, to individualize the test assignments for every student. This is the subject of the current paper.

STUDENT MODELLING AND ADAPTIVE TESTING: A MARRIAGE OF CONVENIENCE?

Student modelling

Student models are usually seen as a label proving intelligence in computer-assisted instruction. The enormous difficulties in creating a precise machine representation of the student's mental state have recently caused a relative lowering of the enthusiasm about intelligent tutoring systems (ITS). Some researchers now even reject entirely the benefits of student models. We believe this is too extreme. Student models are difficult to build, yes, but this is not a reason for denying that they are very valuable for all systems that pursue some degree of adaptivity to the individual student.

In our opinion, when creating an adaptive system, a solution can always be found if the student model is designed in such a way as to serve the specific needs of the application. This will, of course, be a compromise with the unattainable goal of knowing exactly what is going on in the student's head. However, even if we had a way to know that, it would be very difficult to find out how to use this knowledge to improve the interactions with the student (Self, 1988).

Hence, first we have to decide how we are going to use the student model and then design it: that means, to find what information will be represented in it, how it will be represented and how it will be updated.

Adaptive testing

Traditionally (Eskenasi *et al,* 1989), adaptive testing means the selection of the next item with a weight depending on the success of the previous items. Here we shall use a broader notion of adaptive testing. We define three levels at which a testing system can adapt itself to the student:

- deciding what to test the student about;
- selecting an item with appropriate difficulty;
- finding a suitable way to ask the question.

To carry out adaptivity on the first level, the system needs a representation of the knowledge elements it is testing and the links between them. It also needs a representation of the elements that the student already knows (a model of the student's knowledge).

Adaptivity on the third level can be obtained by considering the individual psychological features of the student, so we need a model of the student's individual characteristics. To ensure adaptivity on the second level, the system needs a way to estimate the weight of the items and the capabilities of the student. The capabilities are a function of knowledge and psychological features and can therefore be estimated by using the information of the two student models mentioned above. After a review of the existing student modelling techniques, an appropriate way of representation and updating were chosen for these two models (Vassileva, 1990).

The skill or concept-lattice

An adaptive testing system needs to know what are the skills it is going to test and the links between them. This knowledge can be represented with an *and/or* graph, in which the skills are the nodes and the arcs correspond to the precedence- or logical-links between them. For example, one possible graph in the domain of 'decomposition into partial fractions' can look as shown in Figure 1.

An *and*-link between the children of a node means that all the skills corresponding to the children-nodes need to be mastered (achieved) by the student to master the skill corresponding to the parent-node. An *or*-link means that mastering of the children-nodes is enough to master the parent.

Building a skill or concept-lattice for a given domain is not a trivial task. Two methods have been proposed: an empirical one (Doignon and Flamagne, 1985) and a theoretical one (Kohnert and Lemke, 1990). The skill- or concept-lattice is helpful in selecting which is the next skill to be tested. For example, if the student model shows that the student does not know one of the children-nodes connected with an *and*-link, there is no point in testing on the parent-node. In this way students will be tested more economically (without asking questions they are not going to be able to answer).

THE TEST-ITEM BANK

The multiple choice test-items can be associated with the skills or concepts in the lattice, described above. The distracters are generated to demonstrate lack of knowledge in certain sets of skills. For example, let the item be:

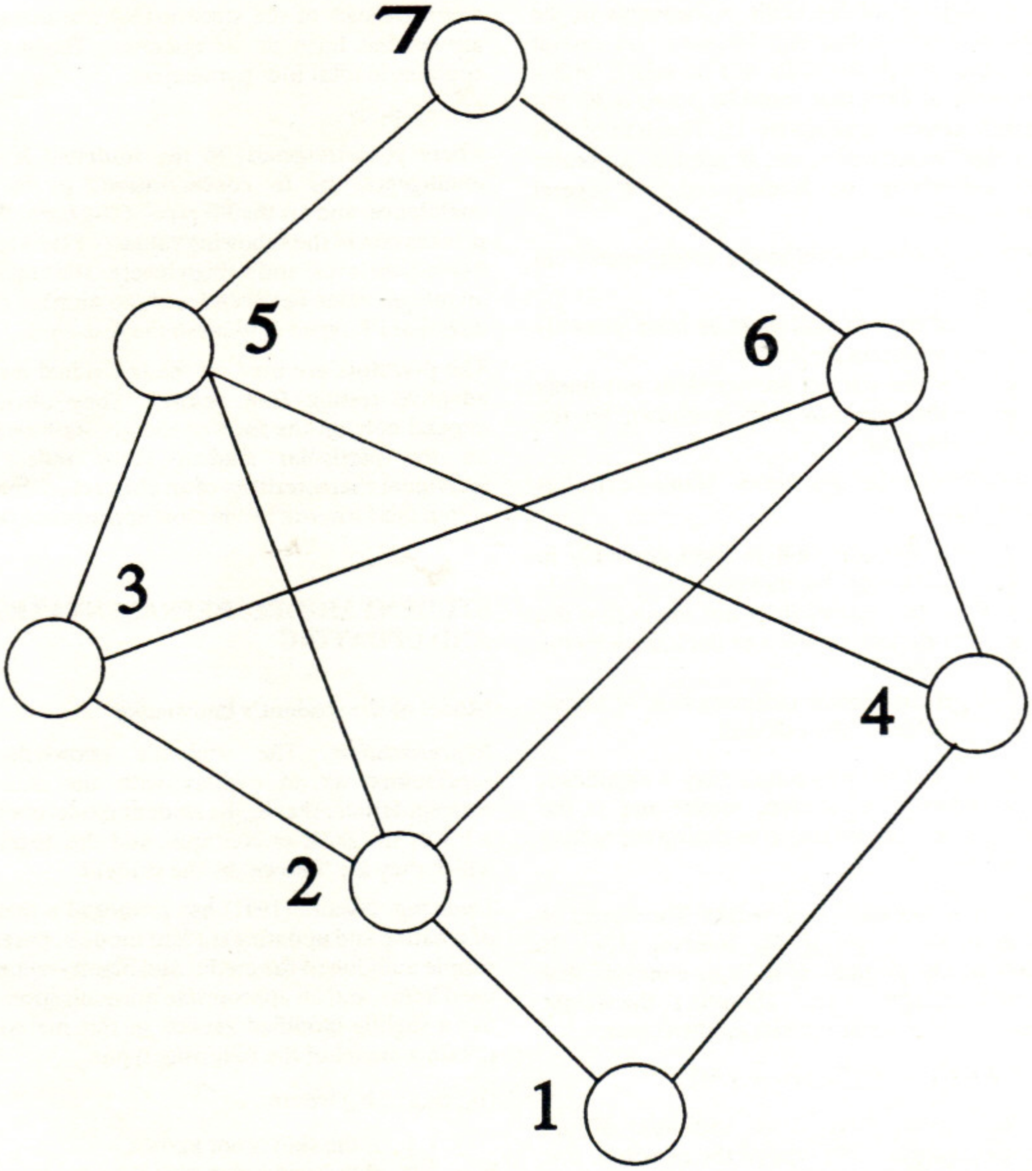
Decompose to partial fractions $(2x+1) / (x+1)^3$

a. $-1/(x+1)^3$
b. $2/(x+1)^2 - 1/(x+1)^3$
c. $A/(x+1)^3 + B/(x+1)^2 + C/(x+1)$.

Here, b. is the correct answer and a. and c. are distracters. The distracter a. indicates that the student does not know that if the denominator is of the type $(x - a)^n$, there should be partial fractions with denominators $(x - a)^{n-1}$, $(x - a)^{n-2}$, ..., $(x - a)$ in the decomposition. The distracter c., where $A<>-1$, $B<>2$ and $C<>0$, shows that the student has not calculated the coefficients correctly, so probably does not know the method of the undetermined coefficients etc.

Every answer, both correct and incorrect, gives evidence for the student's knowledge or absence of knowledge of a certain skill or set of skills (Nwana, 1991). For this reason every answer (correct and incorrect) of an item is associated with a vector:

$(a_1, a_2, \ldots, a_n)$
where n = the number of skills or concepts in the lattice,
a_i indicates whether the i-th skill participates in obtaining the answer.



1. the student can factorize algebraical expression. The student can solve problems where the denominator is of type:

2. $(x-d)(x-b)$
3. $(x-a)^n$
4. $(ax^2+bx+c)^n$; $b^2-4ac<0$
5. $(x-a)^n(ax^2+bx+c)$
6. $(x-a)(ax^2+bx+c)^n$
7. $(x-a)^n(ax^2+bx+c)^m$

Figure 1. *Example of a skill-lattice*

The number 'n' of the skills or concepts in the lattice will not be too big, because one cannot expect that too many skills will be tested with a single item. A limit that could be posed is 30, but a realistic number is about 10–15. The limit of 30 is not a rigid constraint since, if needed, a greater lattice can always be decomposed into several smaller ones.

For the correct answer we have a 'credit-vector', in which:

$$a_i = \begin{cases} 1 & \text{if the i-th skill must be used correctly to obtain the answer} \\ 0 & \text{if the correct answer does not imply that the i-th skill is known by the student} \end{cases}$$

For every incorrect answer a 'blame-vector' is defined as follows:

$$a_i = \begin{cases} 1 & \text{if the i-th skill is used correctly in obtaining the answer} \\ 0 & \text{if the answer does not imply that the i-th skill is known or not known by the student} \\ -1 & \text{if the answer indicates lack of knowledge for the i-th skill} \end{cases}$$

The credit- and blame-vectors play a significant role in updating the student model and in the selection of an appropriate item during the testing process.

Each item has an associated weight. The weight of an item is an integer in the interval [1,w]. It depends on the number of skills or concepts that the item is intended to test. Therefore, the weight of each item is a function of its credit-vector:

$$INTEGER\,\{(w-1)*(\textstyle\sum a_i)/(n+1.5)\}$$

For every item, there is an indicator of the psychopedagogical (PP) type of the question. For example: text; text with a built-in hint; graphics; animation or any combination of these. Each item might exist in up to three forms in the bank and can be issued in any one of them. They are: short, extended and with explanation. These three forms might correspond to different PP-types – for example, if the extended form of the question contains a graphic etc.

Also, for every form of a question there is an associated vector, called the p-vector, containing three parameters corresponding to the individual characteristics of the student that are needed to answer this form of the question. The p-vector contains in total four parameters:

(p_1, p_2, p_3, t)

where p_1 corresponds to the student's level of intelligence, p_2 to concentration, p_3 to self-confidence, and t is the PP-type of the form. Every p_i takes one of the following values: –1 (low level), 0 (medium level) and 1 (high level). The t can take an integer value between 1 and the number of the different PP-types of items in the test-bank.

The p-vectors are used on the individual level of adaptive testing (see below). They obviously depend only on the form of the question and not on the particular student (they reflect the individual characteristics of an abstract student for whom this form will be the most appropriate one).

STUDENT MODEL: REPRESENTATION AND UPDATING

Model of the student's knowledge

Representation. The student's knowledge is represented as an overlay with the skill- or concept-lattice; that is, the student model contains a list of the skills or concepts, and the degree to which they are 'known' by the student.

Updating. Nwana (1991) has proposed a method of creating and updating student models, based on simple addition of the credit- and blame-vectors of used items, and an appropriate normalization. We use a slightly modified version of this method to obtain a model of the following type:

$(b_1, b_2, \ldots, b_n)$ where:

$$b_i = \begin{cases} 1 & \text{if the skill is not known} \\ 0 & \text{if the skill is not tested} \\ 1 & \text{if the student is aware of the concept/skill} \end{cases}$$

Model of the student's individual characteristics

Representation. The model of the student's individual characteristics is represented with a vector of parameters corresponding to three individual features of the student (intelligence, concentration and self-confidence) and the preferred PP-type for the student. The parameters corresponding to the individual characteristics take discrete values: –1 (low), 0 (medium), 1 (high). So, the individual model is:

(m_1, m_2, m_3, t) where

$$m_1 = \begin{cases} -1 \\ 0 \\ 1 \end{cases}$$

and 1<=t<= the number of different PP-types.

The individual model can be initialized with a pre-test based on certain psychological methods (Wittig, 1986). 'Intelligence' can be evaluated on the basis of a general estimation of the student's behaviour (successes) in the subject. The last parameter (t) can be initialized by asking the student to explicitly state the PP-type of preferred items.

Updating. Updating the individual model is a two-fold process. The individual model is used to obtain adaptivity, ie appropriate test items are generated according to the individual characteristics of the student. On the other hand, the student's ability to cope with the selected items provides evidence for his individual characteristics (Vassileva, 1990).

The method of Nwana described above can be used for updating the values of the parameters, corresponding to the student's individual characteristics (m_i). The first three parameters in the p-vectors of the successfully used items are added, and after an appropriate normalization the sums are converted to fit into the [–1,1] interval.

The last parameter (t) in the individual model is updated by generating statistics of the number of successful applications of items of every PP-type. 't' is assigned the number corresponding to the PP-type on which the student has shown the best results.

LEVELS OF ADAPTIVE TEST GENERATION

Global level

The first question that has to be answered in adaptive test generation is what skills or concepts will be tested. If we already have an initial model of the student's knowledge, we can use it to generate a test in which the skills or concepts that are already known will not be tested. For example, let us have a sub-graph in the skill- or concept-lattice (see Figure 2).

Suppose that the lower part of this graph is tested, and the uppermost node is the goal of testing. We can generate a test starting from the left-most not-tested node on level 13 and going up.

Two strategies can be defined:

- Full testing – when all prerequisite-nodes of a goal are tested even if they are connected with an *or*-link. In this way the model of the student's knowledge will contain full information about the degree of knowledge on every skill/concept in the lattice. This strategy of course is not explicitly adaptive.
- Quick testing – when all the prerequisites of a node are tested only if there is an *and*-link between them. This strategy is good if we want to know whether the student has acquired the minimum of knowledge needed for accomplishing the goal of testing, similar to a threshold in adaptive testing (Eskenasi *et al*, 1989) without a student model.

In both strategies it is possible to test the student further after testing the goal-node. Not-tested (*or*-link) nodes or nodes which are tested unsuccessfully before (information about this could be found in the student model) will be tested if there is time, thus letting the student improve his or her results.

Local level

After deciding which skills will be tested, the question arises about which item will be chosen. If a certain skill must be tested, an appropriate item has to be selected from the set of all items in the base that have the value '1' in the corresponding position of their credit-vector.

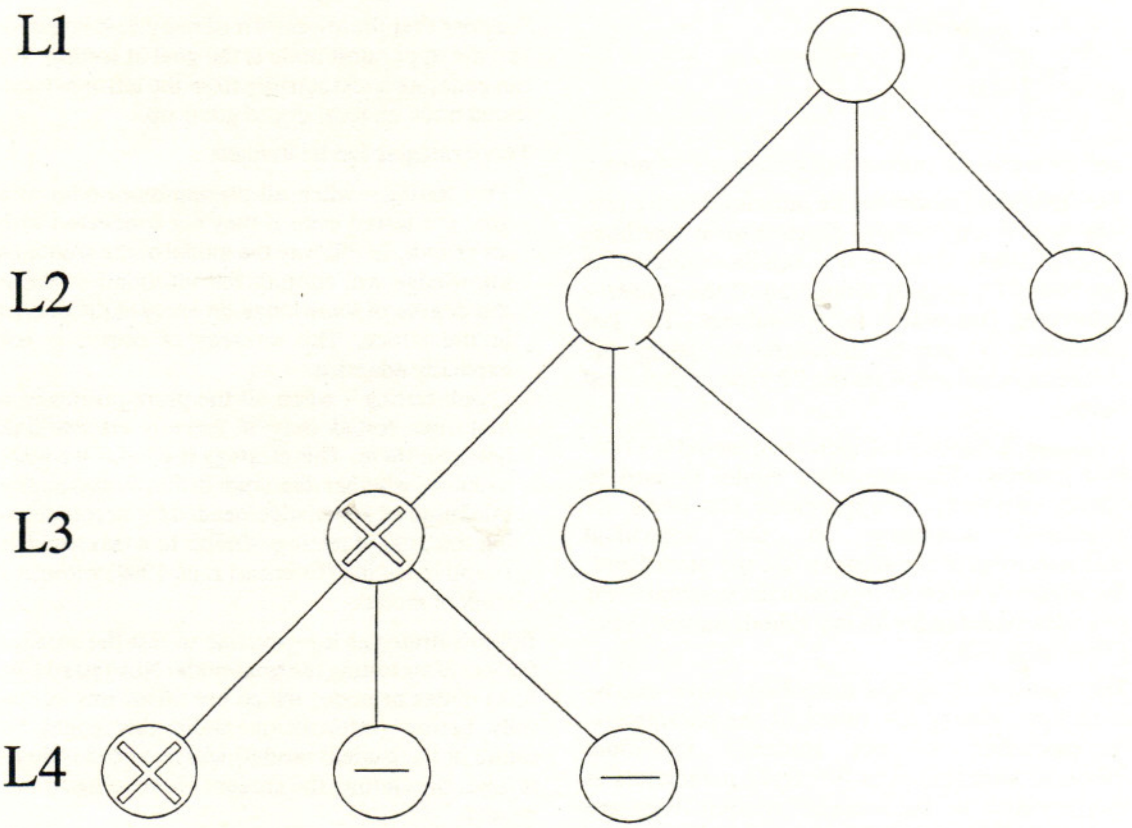
Two strategies are possible:

- *Informative* – choosing an item for which the difference between the credit-vector and the model of the student's knowledge is the maximum. We are seeking item 'k', for which we have:

  $\max \sum |b_{ji} - a_{ji}|$ where

  1<=i<=n n=number of skills or concepts in the lattice

  1<=j, k<=r r=number of items in the test-item bank.



⊗ – successfully tested skill;
⊖ – failure on this skill;
○ – not-tested skill.

Figure 2. *Quick-testing strategy illustration*

In this way we select an item that will test the greatest number of untested or unsuccessfully tested items. Therefore, we obtain more information about the student's knowledge.

- *Adaptive* – choosing an item that is supposed to correspond to the student's knowledge and individual features. In this strategy, the function 'weight' of an item, as previously defined, is used to describe the difficulty of the test item.

We need another function that maps the data from the model of the student's knowledge and his or her individual model into the set of possible weights of items testing a given skill or concept. In our opinion the student's knowledge is the main factor that has to be taken into account; therefore the function can be very similar to that for calculating the weight of an item on the basis of its credit-vector as shown earlier. Here, however, a recommended weight for the student is calculated on the basis of his or her knowledge model. The combined formulae for calculating the weight on the basis of the model of the student's knowledge is:

$$INTEGER\ \{[(w-1) * (\sum b_i + k)/3k] * (h + 1.5)\}$$

where k is the number of tested skills or concepts ($b_i <> 0$) in the student knowledge model. Since

$-k \leq \sum b_i \leq 2k$ (because $-1<=b_i<=2$) the normalization needs a division by 3k (the size of the interval).

A correcting coefficient 'h' should act as a 'filter', reducing or increasing the recommended weight according to the student's individual model. We assume that for a student with higher values of individual characteristics, parameters should be given an item with a higher weight. However, 'h' should not be the main factor. If the student's individual model suggests that he or she is not intelligent and not confident, the student should not be sentenced to receive only items with the minimum weight. That is why 'h' should never become 0. Otherwise, the model of the student's knowledge will not be taken into account and a minimum weight item will always be generated. We decided that 'h' will vary in the interval [0.5, 1]. In this way, students who have high intelligence, concentration and confidence will get an item with a weight calculated directly on the basis of the model of their knowledge (h=1). Other students will get an easier item. 'h' is calculated on the basis of the individual student model by the formulae:

$h = (\sum m_i+3)/(12+0.5)$.

Individual level

When the item is chosen, the form of the question has to be selected. The individual student model is used for this purpose. Two strategies are possible:

- minimizing the difference between the individual student model and the p-vector of the question. We have determined an item with one of the strategies described in the previous section and are looking now for a form of the question of this item, for which we have:

  $\min \sum |p_{ij} - m_{ij}|$ where
  $1 \leq i \leq 3$
  $1 \leq q$
  $j \leq$ number of forms for the selected item.

  In this way an appropriate form of the question of the test-item will be selected for the individual student.
- Selecting a form whose PP-type corresponds to the most successful PP-type for the student. The last parameter of the individual model can be used to select a form of the most successful type for the particular student. A measure of similarity or neighbourhood can be defined between different PP-types. For example, a form containing an animation is close to a form containing static graphics. This measure will be used to select a 'second most appropriate' form of the given item if there is no form of the most successful type for a given item.

With both strategies the student is allowed to ask for another form of the item, if any other forms of the selected item are present.

REFERENCES

Bruno, J.E. (1989) Monitoring the Academic Progress of Low Achieving Students: An Analysis of Right-Wrong vs. Information Referenced Formative and Summative Evaluation Procedures. *Journal of Research and Development in Education,* **22,** 4, 51–61.

Doignon and Flamagne (1985) Spaces for the Assessment of Knowledge. *International Journal of Man-Machine Studies,* **23,** 175–196.

Dosher, and Bruno, J.E. (1981) Simulation of Inner City Testing. *American Educational Research Journal,* **18,** 4, 475–489.

Eskenasi, A., Vassileva, J. and Boyadjieva, T. (1989) Management of the Computer-Assisted Testing Process. *Proceedings of the 3rd Conference 'Children in the Information Age',* Sofia, Bulgaria, pp. 188–199.

Kohnert, K. and Lemke, A. (1990) Knowledge Representation with AND-OR Graphs: Comparing the Approach of Doignon and Flamagne with ABSYNT Diagnosis. In D. Norrie and H.-W. Six (eds) *Computer-Assisted Learning, Springer Lecture Notes in Computer Science,* No 438, pp. 158–174.

Nwana, H. (1991) User Modelling and User Adapted Interaction in an Intelligent Tutoring System. *User Modelling and User Adapted Interaction.* **1,** 1, 1–28.

Self, J. (1988) Bypassing the Intractable Problem of Student Modelling. *Procedures of ITS 88,* Montreal.

Vassileva, J. (1990) A Classification and Synthesis of Student Modelling Techniques in Intelligent Computer-Assisted Instruction. In D. Norrie and

H.-W. Six (eds) *Computer-Assisted Learning, Springer Lecture Notes in Computer Science,* No 438, pp. 202–213.

Wittig, A. (1986) *Psychology of Learning, Shaum's Outline Series,* McGraw-Hill.

BIOGRAPHICAL NOTES

Dr Avram Eskenasi is currently the head of the Software Engineering Department at the Institute of Mathematics (Bulgarian Academy of Sciences). His main interests are in the fields of computer-assisted testing and software quality evaluation. Dr Eskenasi is the Bulgarian representative to IFIP TC 2 on Software.

Mrs Tsvetelina Vladimirova graduated from the University of Sofia in 1990. Presently she is a postgraduate student at the Institute of Mathematics in Sofia. The subject of her research is computerized testing.

Dr Julita Vassileva graduated from the University of Sofia in 1986. In 1991 she received her PhD for a thesis in the field of student modelling. She has already published several articles in foreign journals, submitted and presented papers to international conferences and taken part as a member of programme committees of other scientific events. Presently she is with a research institute in Germany.

Address for correspondence: Software Engineering Department, Institute of Mathematics, Bulgarian Academy of Sciences, IIII Sofia, Bulgaria.