Internet based Evaluation System

A. Ríos, E. Millán, M. Trella, J.L. Pérez-de-la-Cruz and R. Conejo
Departamento de Lenguajes y Ciencias de la Computación
Facultad de Informática, Campus de Teatinos, 29071. Málaga, Spain.
{rios, eva, trella, cruz, conejo}@iaia.lcc.uma.es

Abstract. In this paper, we describe the design and development of a web-based Computerized Adaptive Testing system (CAT) that is still under development and will be one of the main components of the TREE project. The TREE project consists in the development of a several web-based tools for the classification and identification of different European vegetable species (an expert system, interfaces for creating and updating databases and an intelligent tutoring system).

The test generation system will be used by the ITS diagnostic module, and has a complete set of tools that not only assists teachers in test development and design, but also supports student evaluations. Adaptive capabilities are provided by an IRT model. While the student is taking the test, the system creates (and updates) his/her *temporary student model.* In this way, the system can be used in two different ways: as an *independent evaluation tool* over the WWW (SIETTE system, already finished), or as a component of the diagnostic module in any ITS with a *curriculum structured* knowledge base as the TREE ITS.

Key words: evaluation system, authoring tools, adaptive testing, student model, diagnosis.

1 Introduction

In this paper, we will describe one of the main components of the TREE project. (TRaining of European Environmental trainers and technicians in order to disseminate multinational skills between European countries). The TREE project is included in the EU Leonardo da Vinci Program, and its main goal is the development of an ITS for the classification and identification of different European vegetable species.

The main modules of the tool being developed in the TREE project are: *the Expert System* (ES), the *Intelligent Tutoring System* (ITS) and the *Test Generation System* (TGS), as shown in Figure 1.

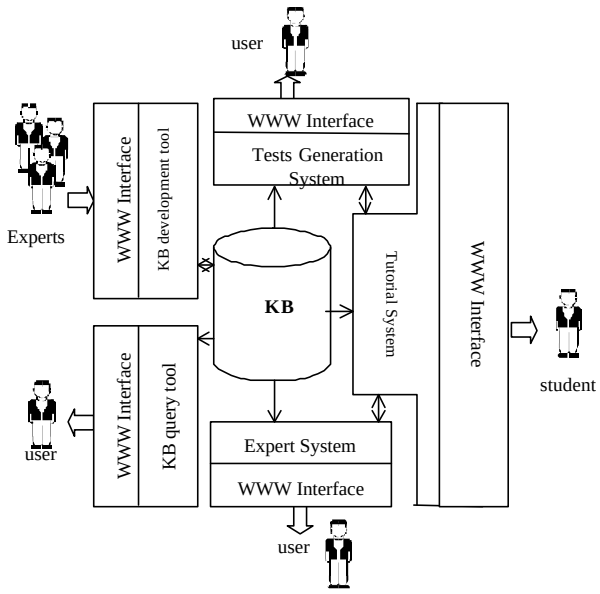

Figure 1. Structure of TREE.

All these tools make use of the *Knowledge Base* (KB) that contains all the information about the botanical domain and is being incrementally built by different users in different

locations. Each one of the components, including the knowledge base, has an independent web-based interface that allows the whole system to be used as a learning tool or as an independent consultation tool.

The test generation system used in TREE has been developed and implemented as an independent and reusable system for the design and generation of adaptive tests over the WWW. Also, this system can interact with any ITS that has:

- a curriculum structured knowledge base, and
- a student model defined as an semantic network, where each node is a curriculum component with an associated knowledge level.

The system described in this paper tries to join the dynamic nature of computer adaptive tests and the advantages that the WWW offers as learning environment (multimedia content, hypertext capabilities, client/server architecture). In this way, evaluation methods can be included in distance learning systems to provide them with adaptive capabilities.

2 Adaptive Testing

In traditional paper and pencil test evaluation methods, the storage and analysis of the information is static. The use of computers in testing processes opened up the possibility of making these processes dynamic. An *computer adaptive test* is a computer administered test where the presentation of each item and the decision to finish the test are dynamically adopted based on the student answers, and therefore based on his/her proficiency. By using a temporary student model, an adaptive test can also generate descriptive information about the student learning style and problems.

The theory used to provide computerized adaptive tests with adaptive capabilities is *Item Response Theory* (IRT). IRT is a statistical framework in which examinees can be described by a set of ability scores that are predictive, linking actual performance on test items, item statistics and examinee abilities. The two central issues in adaptive testing are:

a) *Calibration of items*. In order to provide computerized testing systems with adaptive capabilities, questions have to be calibrated with some parameters. These parameters will guide the question selection strategy.
b) *Content-balance*. The test has to cover all the important content areas in the subject.

These important issues are discussed in detail in [1].

The main advantages of adaptive testing over traditional testing are:

- A significant decrease in the test length (consequently, in the testing time), as shown in the comparative study reported in [2].
- More accurate estimations of student knowledge level. Besides, usually this information is more detailed and therefore a better use of it can be made when trying to provide feedback or take instructional decisions.

Our web-based tool to assist in the evaluation process is very simple to use, and makes these advantages accessible to educators all over the world. Moreover, this tool can help in the development of complete educational web-based systems that not only deliver the instructional material, but also provide detailed performance measurements relative to the learning process. This information that can be used either by instructors (evaluation), Intelligent tutoring systems, (instructional decisions) or by students (self-evaluation).

3 The SIETTE System

The web-based tool we have developed is called SIETTE (Intelligent Evaluation System using Tests for Teleeducation; [3]). SIETTE can be used in two different ways:

- Instructors and domain experts can use SIETTE to define tests,
- Students can use SIETTE to take the tests that are automatically generated according to the specifications provided by instructors and the information stored in the temporary student model.

Once that the students have taken the tests, the system presents detailed information about the performance of each student.

The architecture of the SIETTE system is shown in Figure 2. In order to give a general overview of the system, we will briefly describe all the components and then, we will discuss the most important ones in more detail.

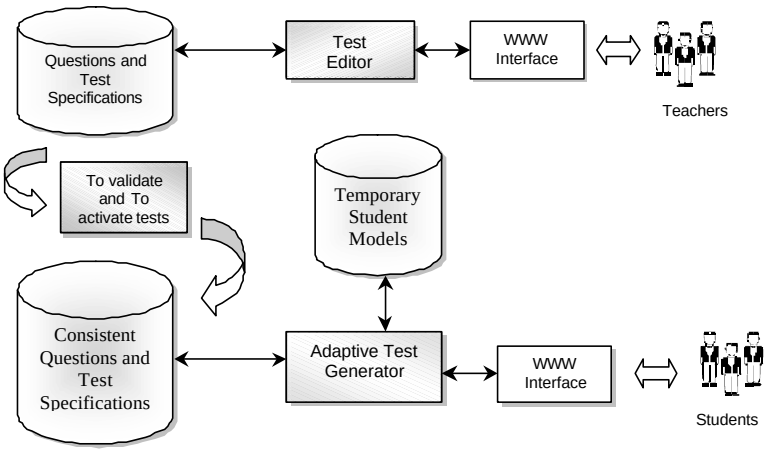

Figure 2. SIETTE Architecture

- The *question knowledge base* is a collection of possible questions to pose in a test. All these questions are calibrated with some parameters.
- The *test edition module* is the tool used by instructors or domain experts to define the tests. This module allows to define the structure of the domain: topics, questions, relations between them and relative weights of topics in the test. This information is stored in the question knowledge base. Moreover, in this module the test developer can define *test specifications* that will guide the question selection process and the finalization criteria, such as maximum number of questions to be posed, minimum number of questions of each topic, degree of confidence in the estimated knowledge level, etc.
- Once the tests have been defined, there is a module that validates their elements (topics, questions, specifications) and activates the tests so they can be used by the adaptive test generation system. This test validation and activation module is an off-line process that is run in the server side, where the database server is.
- A *temporary student model*, that is created and updated by SIETTE for each student that takes the test. Basically, it consists in a vector of eleven probabilities (p_0, p_1, ..., p_{10}), where p_i represents the probability that the student has reached a knowledge level i, and also information about which questions have been asked by the system.
- The *test generator* is the main module of SIETTE. It is responsible of selecting the questions that will be posed to each student. The generation process will be guided by the *specifications* defined by test developers and by the *temporary model* of the student that is taking the test.

Specific interfaces have been implemented to make test edition and test generator modules accessible via WWW. Using these interfaces, it is possible to add questions, answers and test specifications to the knowledge base, and also to modify them. The knowledge base has been implemented using a relational database that can be accessed via WWW with scripts.

Now we will describe the test editor, the temporary student model and the test generator in more detail.

TEST EDITOR

The test editor is a tool that uses HTML forms to extract knowledge from domain experts or instructors. The information supplied is saved in a relational database so it can be used by the test generator module.

In Figure 3 we can see the interface for the test editor:

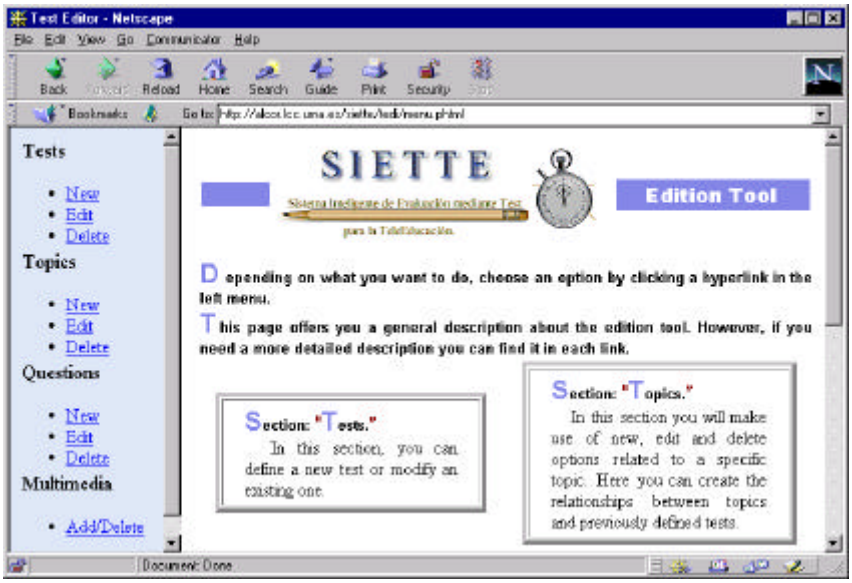

Figure 3. Test editor interface

As we see, in SIETTE, tests have a curriculum-based structure. Each test for a particular subject is structured in *topics* and *questions*. The editor also allows the definition of relationships between tests, topics and questions. The curriculum defined in this way will be used by the question selection algorithm to generate content-balanced tests adjusted to the specifications previously defined, as described in [4], [5] and [1]. Besides, test designers can define parameters associated to topics and questions: a *weight* for each topic that represents how important is the topic in the test, and a *degree of difficulty* for each question. This degree of difficulty will be then combined with the *guessing factor* (1/number of possible answers) and the *discrimination index* to construct the *item characteristic curve* (ICC) associated to each question. In IRT, the ICC is given by the probability that a student answers the question in the right way given his/her current knowledge level. This procedure eliminates the need of empirical previous studies used by IRT to calibrate questions [6]. Finally, using the test editor, it is possible to add multimedia content (graphics, video, sound) to questions and answers. By including multimedia contents, a greater number of subjects can be evaluated using SIETTE. The mechanism for storing multimedia content via WWW is based on RFC 1867 (Form-based file upload in HTML) [7].

Another important area developed in the SIETTE system is the possibility to define *question and answer templates.* These templates will be dynamically instantiated if the item (question or answer) is selected, and the system will randomly choose one of such instantiations.

So the main advantages that this edition module offers are;

- *component reusability*: tests for the same subject can share topics, and these can share questions.
- Test components are written using HTML, with the flexibility that this language offers.
- *multimedia content* in questions and possible answers,
- by using *templates*, a great number of different questions (or answers) can be automatically generated by the system.

THE TEMPORARY STUDENT MODEL

The temporary student model is created and updated by the system for each student that takes the test. This information will be used to provide the test generator module with adaptive capabilities.

Currently, the student can be classified into 11 knowledge levels, ranging from Level 0 (novice) to Level 10 (expert). Initially, the probability is uniformely distributed between the 11 levels. As the student takes the test, these probabilities are updated with a bayesian procedure. In Figure 4 we can see an example of a temporary student model and how it is structured:

STUDENTS					
StudentID	TestID	Date	Level of Proficiency	Lower Confidence Level	Upper Confidence Level
John Smith	TREE	04/08/98	Level 1	0.9	1.1

KNOWLEDGE DISTRIBUTION					
StudentID	TestID	Level 0	Level 1	...	Level 10
John Smith	TREE	0.001	0.9	...	0.001

TOPIC DISTRIBUTION			
StudentID	TestID	TopicID	% Questions
John Smith	TREE	PINUS	40%
John Smith	TREE	ABIES	40%
John Smith	TREE	CEDRUS	20%

QUESTIONS POSED		
StudentID	QuestionID	AnswerID
John Smith	Q_1	$A_{1,1}$
John Smith	Q_3	$A_{3,2}$
John Smith	Q_5	$A_{5,1}$
John Smith	Q_6	$A_{6,3}$
John Smith	Q_8	$A_{8,4}$
John Smith	Q_{10}	$A_{10,5}$
John Smith	Q_{12}	$A_{12,3}$
John Smith	Q_{14}	$A_{14,1}$
John Smith	Q_{17}	$A_{17,2}$
John Smith	Q_{20}	$A_{20,1}$

Figure 4. An example of a temporary student model.

TEST GENERATOR AND EVALUATION ALGORITHM

To generate the test we use Owen´s Bayesian approach as described in [8]. Mainly the procedure works by calculating the posterior probabilities of a student having a certain knowledge after he/she gives an answer a to question n. However, instead of considering student's knowledge as a continuous random variable we have consider it as a discrete random variable whose possible values are {0,...,10}. This assumption simplifies computations needed for the estimation of the new knowledge level and its confidence interval. This module has been implemented using a CGI application. The test generator algorithm consists of three procedures:

1 Question selection

Test developers can choose between three different question selection procedures:

- *Bayesian procedure*: (selecting the item that minimizes the posterior standard deviation),
- *Adaptive procedure*: (selecting the item which gives the minimum distance between the mean of the ICC and the mean of current student model),
- *Random procedure*: (the item is selected randomly).

Whatever procedure is used, the system extends Owen's approach with these features:

- *Random item selection.* If the selection criterium does not allows to differentiate between two questions a random selection is done. This usually happens when using templates, because ICCs are assigned to templates, so every instance of a template has the same ICC.
- *Content Balancing*. In SIETTE, student's knowledge is represented by a variable θ, that is, SIETTE uses an unidimensional model. However, to assure content balanced tests, SIETTE uses the weights specified by the test designer for each topic included in the matter being evaluated. These weights determine the desired percentage of questions about each topic. SIETTE compares the empirical percentages of the questions that have already been posed with the desired percentages, and selects the topic with the biggest difference as the target topic. Then, SIETTE selects the best next question belonging to that topic using the ICC associated to each question.

- *Longitudinal testing*. Item selection strategy in SIETTE avoids posing the same items to a student who takes the test more than once. The selection strategy uses the information stored in the student model about items posed in earlier tests.

2 Updating the temporary student model

Once the best question has been chosen, the system poses it to the student and waits for an answer. When the student answers the question, SIETTE system computes his/her new proficiency level and its confidence interval. With the new proficiency level, the confidence interval, and information about questions posed and coverage of the test, the system updates the temporary student model.

3 Termination criterion

The termination criterion can also be determined by test developers, and it can be any valid combination (using OR) of the cases listed below:

(1) The standard deviation of the distribution of the student knowledge is smaller than a fixed value (the estimation is accurate enough).
(2) The probability that the student knowledge is greater or equal than a fixed proficiency level is greater than a fixed number.
(3) The system has already posed all the questions in a test.
(4) The system has posed at least the minimum number of questions of each topic specified by the test designer.

Once the test finishes, the temporary model of each student becomes the *student model* of the examinee.

One of the characteristic of the system is the capability to offer immediate feedback. While a student is taking a test, the system can show him/her the solution to the question she has just answered. In this moment, the student could try to cheat the system by pressing the navigator BACK button. To avoid this type of behavior, we have implemented a simple state machine that is shown in Figure 5:

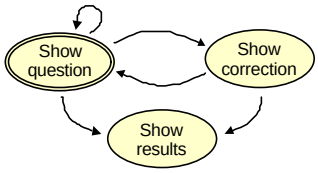

Figure 5. State machine for avoiding cheating.

4 An example

In this section, we will present an example. Let us suppose that a new student is going to take a TREE test (about a botanic domain). In the TREE test specifications, the minimum level of knowledge required in order to pass the test is 5, and the level of confidence is 75%.

Initialization of the temporary student model

Initially, and in absence of any type of information, the knowledge level of the student is consider to be a uniform distribution, that is, the probability that the knowledge level of the student is i (for i = 0,1...,10) is 1/11, as shown in the first window in Figure 6

Selection of the first question

First the algorithm selects the target topic, that is the one with the biggest weight in the test (the most important topic). Then it selects one of the questions belonging to that topic, using the ICC for that question. The question selected is the one that minimizes the a-posteriori variance. In Figure 6 we can see the first question selected, and its associated ICC.

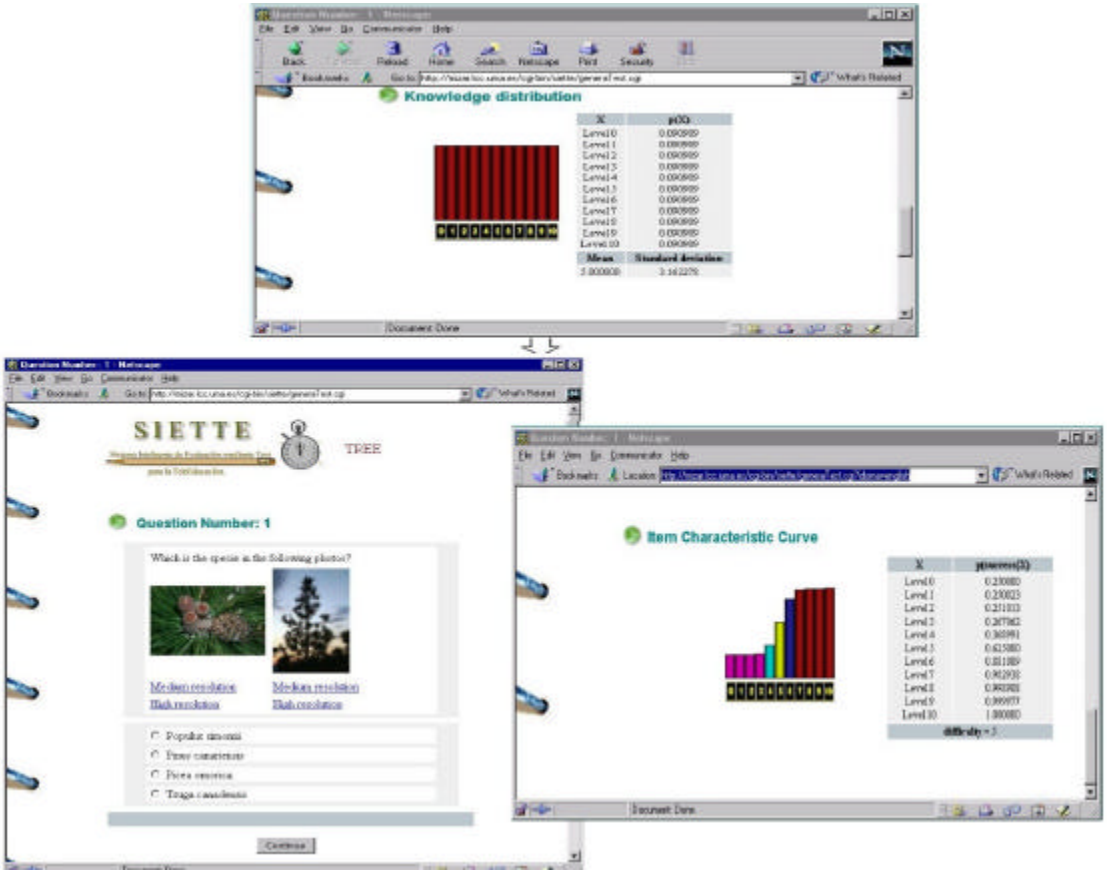

Figure 6. Initial state in a test session, first question presented and its ICC

Now the student will answer the question, and the system will update the temporary student model and use it to select next question. In Figure 7, we show an intermediate state after the student has already answered seven questions:

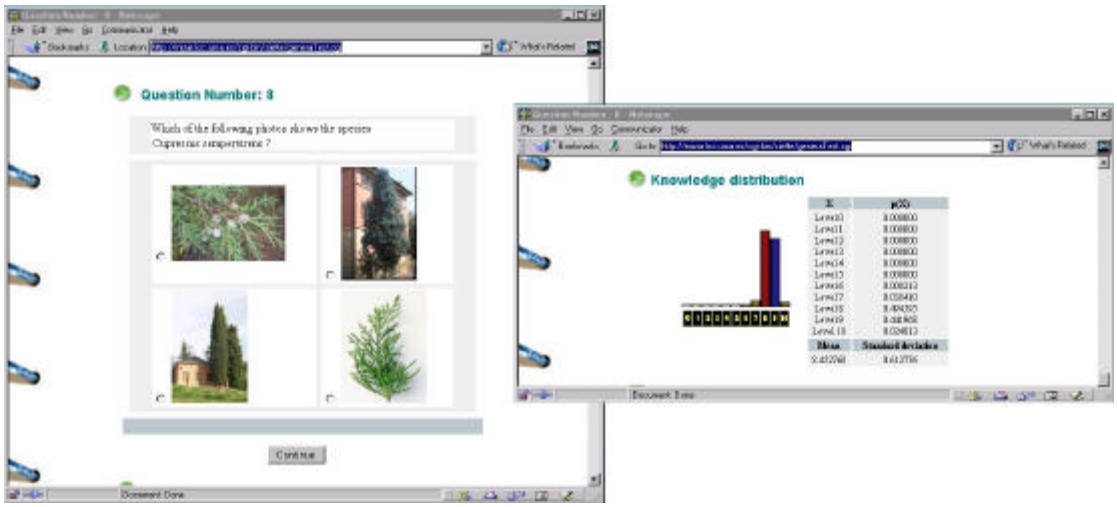

Figure 7. Question 8 and knowledge distribution after seven questions

Now the probability that the student´s knowledge level is 8 is 0.49. The test goes on, and after 11 questions it finishes. The final result is shown in Figure 8: The student knowledge level is estimated to be level 9 with a probability of 0.823070, so the test finishes and the student's final estimated knowledge level is 9. We can also see the statistics that SIETTE presents when the test has finished: number of posed questions, number of correct answers, estimated knowledge level and confidence interval for this estimation. As the level of knowledge reached by the student is 9, according to the test specifications the student has passed the test.

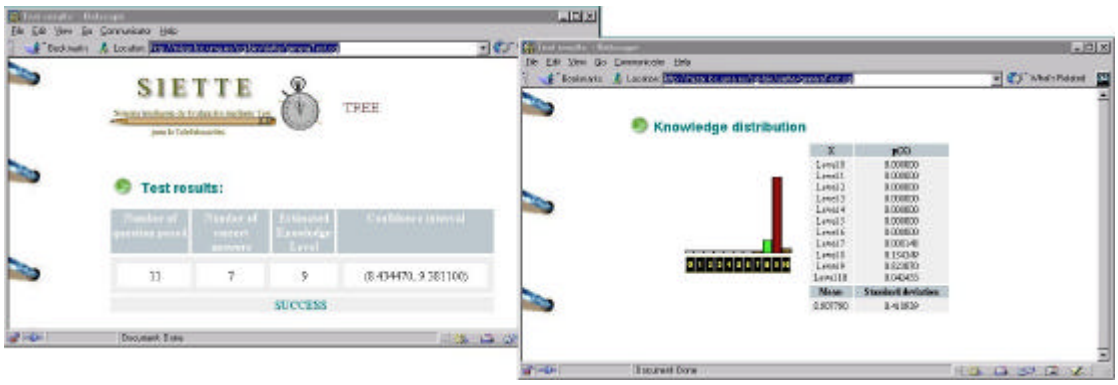

Figure 8. Final test session view

5 Conclusions and future work

The system we have implemented combines the dynamic nature of computerized adaptive testing systems with the advantages that the WWW offers as learning environment. Using the WWW we can help teachers all over the world in the difficult task of evaluation. With our system, evaluation is impartial and the results are more consistent and accurate than with traditional paper-and-pencil tests. All the tools that compose the SIETTE system can be accessed simultaneously, so many different persons can use the system with different purposes at the same time. SIETTE uses HTML-like language for editing questions, therefore both format and aspect of questions are totally adaptable to teacher's preferences. Generated tests can contain multimedia objects, so teachers can compose more attractive test interfaces and evaluate subjects (for example, all subjects that involve recognition of objects from photographs) that cannot be evaluated using only text.

Finally, we would like to remark the importance of using efficient algorithms to select the best question to ask and also to store and recover the information in the student model. Delay times due to these processes and also to downloading ones can keep students waiting for too long when using the system. To improve the average performance, it may be interesting to use the Internet time delay to run the algorithms selecting the next question in the server side, while the student is still waiting or thinking about the last question in the client side. Also, in further improvements of the system we want to include polytomous items and also to use multidimensional models so we can have a more detailed information in the student model to be used by the ITS component of the TREE Project.

REFERENCES

[1] Huang, Sherman X. (1996). On Content-Balanced Adaptive Testing. *LNCS 1108*, 569-577.

[2] Collins, J.A., Greer, J.E. and Huang, S.H. (1996). Adaptive Assessment Using Granularity Hierarchies and Bayesian Nets. *LNCS 1086*, 569-577.

[3] Ríos, A. M., Pérez de la Cruz, J. L. and Conejo, R (1998). SIETTE: Intelligent Evaluation System using Tests for TeleEducation. Workshop "Intelligent Tutoring System on the Web" at ITS'98. http://www-aml.cs.umass.edu/~stern/its98/

[4] Kingsbury and Zara (1989). Procedures for Selecting Items for Computerized Adaptive Tests. *Applied Measurement in Education*, 2(4), 359-375.

[5] Welch, R. and Frick, T. W. (1993). Computerized adaptive testing in instructional settings. *Educational Technology Research and Development*, 41, 47-62.

[6] Weiss D. J. and Kingsbury G (1979). An Adaptive Testing Strategy for Mastery Decision. *Journal of Educational Measurement.* 21(4), 361-375.

[7] Nebel, E. and Masinter, L (1995). RFC 1867: Form based File Upload in HTML. http://sunsite.auc.dk/RFC/rfc/rfc1867.html

[8] Owen, R.J. (1975). A Bayesian sequential procedures for quantal response in the context of adaptive mental testing. *Journal of the American Statistical Association* 70, 351-356.