International Journal of Artificial Intelligence in Education 14 (2004) 1 -33
IOS Press

SIETTE: A Web–Based Tool for Adaptive Testing

Ricardo Conejo, Eduardo Guzmán, Eva Millán, Mónica Trella, José Luis Pérez-De-La-Cruz & Antonia Ríos, *Departamento de Lenguajes y Ciencias de la Computación, Facultad de Informática, Campus de Teatinos, 29071. Málaga, Spain.*
conejo,guzman,eva,trella,perez@lcc.uma.es

Abstract. Student assessment is a very important issue in educational settings. The goal of this work is to develop a web-based tool to assist teachers and instructors in the assessment process. Our system is called SIETTE, and its theoretical bases are Computer Adaptive Testing and Item Response Theory. With SIETTE, teachers worldwide can define their tests, and their students can take these tests on-line. The tests are generated according to teachers' specifications and are adaptive, that is, the questions are selected intelligently to fit the student's level of knowledge. In this way, we obtain more accurate estimations of student's knowledge with significantly shorter tests. By using the computer, larger question databases can be stored, selection algorithms can be performed efficiently, and questions can include multimedia content. The use of Java applets allows the inclusion of executable content in question stem and/or answers, so the student can interact with the system by means of this applet. In this way, new possibilities are added to Computer Adaptive Tests, such as using traditional multiple-choice questions together with questions whose answer is evaluated by the applet itself.

INTRODUCTION

Assessment has always been a very important step in the learning process. Its different forms are motivated by its different purposes: *exams*, so teachers can know if a student has reached the appropriate level of knowledge; *self-assessment*, for students to check how much they are learning; *questions*, so that teachers can provide the proper type of feedback while teaching, etc.

With the advent of computers, new possibilities were opened up in the field of assessment, e.g., *Computer Adaptive Tests* (CATs). A CAT is a test administered by a computer, where the selection of the next question to ask and the decision to stop the test are performed dynamically based on a student profile which is created and updated during the interaction with the system. The main difference between CATs and traditional *Paper and Pencil Tests* (PPTs) is the same difference that exists between traditional Training Systems and Intelligent Tutoring Systems, that is, the *capability to adapt* to each individual student. In this way, a CAT allows us to go back to a time when teachers could afford to evaluate each student orally by asking him/her a few well-selected questions that were progressively more difficult for students with higher levels (to allow them to show how much they had learned), or easier for students with lower levels (so the teacher

could quickly diagnose that they had not learned enough). As the number of students in the classrooms grew larger and larger, teachers were forced to evaluate their students using standard exams almost exclusively. The problem with standard exams is that, in order to be able to discriminate between all the different knowledge levels, they have to include questions at all levels of difficulty. As a result, tests are longer and include questions that are not *informative* for some of the students taking them, either because they are too difficult or too easy.

The advantages of CATs have widely been discussed in the literature (Kingsbury & Weiss, 1983), and more recently reported by Wainer, Dorans, Flaugher and other authors in (Wainer, 1990). The main advantage is a significant decrease in test length, with equal or better estimations of the student's knowledge level. This advantage is a direct consequence of using adaptive item selection algorithms, that is, algorithms that choose the best (most informative) question to ask next, given the current estimation of the student's knowledge. Some other advantages come from using a computer to perform the tests: larger databases of questions can be stored, selection algorithms can be performed efficiently, and a greater number of students can take the tests at the same time, even if they are in different geographical locations.

The psychometric theory underlying most CATs is *Item Response Theory* (IRT). In IRT, it is assumed that the knowledge level of the student is measured with a single variable θ that is called the *trait*. Using as input data a set of responses of the students to a set of questions, the level of knowledge of the student is estimated (with some statistical method). Then, this estimation $\hat{\theta}$ is used to determine the most informative item to ask next. These steps are repeated until some stopping criterion is met. Different statistical methods to estimate θ and to select the next best question to ask give different IRT models.

The goal of our research is to develop a web-based tool to assist in the assessment process. In this way, we intend to make the advantages of CATs readily available worldwide, and to allow teachers to define their tests and evaluate precisely their students with a minimum of effort. We have called this tool SIETTE (Ríos, Conejo,Trella, Millán & Pérez-de-la-Cruz,1999), (that stands for the Spanish translation of Intelligent Evaluation System using Tests for TeleEducation). The tool that we have developed can be used in two different ways: as a test editor, so educators can define their tests in an easy way, and as an assessment tool, so students can take the tests (previously defined by their educators) online.

In the next section we will briefly review the theoretical background underlying the SIETTE system, that is, Computer Adaptive Testing and IRT theory. Then we will describe in detail the SIETTE system and its capabilities, present an example, and offer some conclusions achieved by our research.

THEORETICAL BACKGROUND

In this section we will introduce Computer Adaptive Testing and IRT theory, which are the basic theoretical concepts underlying the SIETTE system. An excellent primer to CATs and IRT can be found in (Rudner, 1998), where it is possible to try an actual CAT online. For a more detailed description, see (Wainer, 1990) and (Van der Linden & Hambleton, 1997).

Computer Adaptive Tests

Multiple-choice tests are a widely used technique for assessment in the education field. Traditional design methods depended to a great extent on the individual or collective nature of the tests. The tests administered to groups are less expensive in time and resources than those created specifically for a single person, and they have the advantage that all the examinees take the tests under the same conditions. However, in order to be able to evaluate all the examinees[1] precisely (that is, in order to be able to discriminate between them), these tests must contain items with as many difficulty levels as knowledge levels exist in the group of examinees, whereas individually designed tests can contain items chosen more appropriately. A non-desirable consequence is that examinees with higher levels can be bored if they are asked questions that are too easy for them, or the ones with lower levels can be discouraged by trying to answer too difficult questions.

At the beginning of the 1970s, some works pointed out that the creation of more flexible tests could solve some of these problems. In (Lord, 1970), the theoretical structure of an adaptive test was defined, so it could be administered to groups, but tailored to each individual. The philosophy underlying an adaptive test is well described in (Wainer & Mislevy, 1990) *"The basic notion of an adaptive test is to mimic automatically what a wise examiner would do*", that is, if an examiner poses a question that turns out to be too difficult, then the next one should be easier. However, trying out adaptive tests seriously was not possible until the beginning of the 1980s, when computers became more powerful and less expensive. A *Computer Adaptive Test* can be defined as a test administered by a computer where the presentation of each item and the decision to finish the test are dynamically adopted based on the examinee's answers, and therefore based on his/her proficiency. In more precise terms, a CAT is an iterative algorithm that starts with an initial estimation of the examinee's proficiency level and has the following steps:

1. All the questions in the database (that have not been administered yet) are examined to determine which will be the best to ask next according to the current estimation of the examinee's level.
2. The question is asked, and the examinee responds.
3. According to the answer, a new estimation of the proficiency level is computed.
4. Steps 1 to 3 are repeated until the stopping criterion defined is met.

This procedure is illustrated in Figure 1.

1 In this section we will use the word examinee instead of student, following the traditional terminology in Testing Theory.

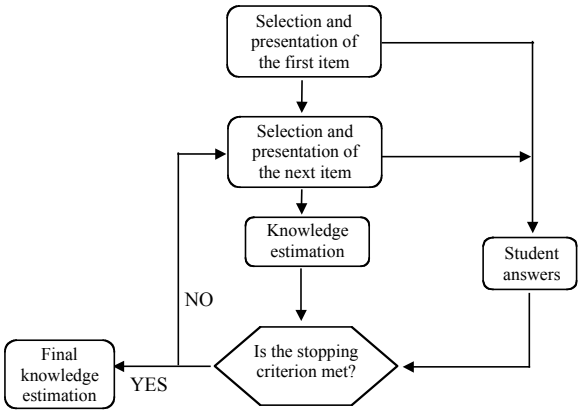

Fig. 1. Flow diagram of an adaptive test. Adapted from (Olea & Ponsoda, 1996).

In this way, the basic elements in the development of a CAT are:

- The response model associated to each question. This model describes how examinees answer the item depending on their level of ability. When measuring the proficiency, the result obtained should be independent of the tool used, that is, this measurement should be invariant with regard to the sort of test and to the individual that takes the test.
- The *item pool*. This is one of the most important elements in a CAT. A good item pool must contain a large number of correctly calibrated items at each ability level (Flaugher, 1990). Obviously, the better the quality of the item pool, the better the job that the CAT can do.
- *The input proficiency level.* Choosing the difficulty level of the first question in a test suitably can considerably reduce the length of the test. Different criteria can be used, for example, taking the average level of examinees that have taken the test previously or creating a examinee profile and using the average level of examinees with a similar profile (Thissen & Mislevy, 1990).
- *Item selection method.* Adaptive tests select the next item to be posed depending on the estimated proficiency level of the examinee (obtained from the answers to items previously administered). Selecting the best item to ask given the proficiency level estimated can improve accuracy and reduce test length.
- *The termination criterion.* Different criteria can be used to decide when the test should finish, depending on the purpose of the test. An adaptive test can finish when a target measurement precision has been achieved, when a fixed number of items has been presented, when the time has finished, etc.

To conclude with this short presentation of the basics of CATs, we would like to address the advantages that CATs have over traditional PPTs (Mislevy & Almond, 1997).

- A significant decrease in test lengths (consequently, in testing time),

- more accurate estimations of examinees' knowledge level,
- an improvement in examinees' motivation,
- large item pools can be stored and managed,
- items can include multimedia content that is more difficult and expensive to use in traditional PPTs, like sound, video, high-quality images, etc.,
- selection algorithms can be performed efficiently.

Item Response Theory

Most of the practical applications of the Theory of the Measurement in Psychology and Education are based in the *Classical Test Theory,* developed between 1920 and 1940. Deficiencies of this theory encouraged the search of alternative models. One of the more relevant ones is *Item Response Theory* (IRT) (Birnbaum, 1968) and (Hambleton, 1989), initially known as *Latent Trait Theory*. IRT, based on strong hypotheses, tries to give probabilistic foundations to the non-observable trait measurement problem. Its name is due to considering the items as the basic units of the test, in contrast to Classical Test Theory that is based on norm-referenced testing.

All IRT-based models have some common features: (1) they assume the existence of latent traits or aptitudes (in our particular case the trait is the examinee's knowledge level) that allow us to predict or explain examinee behavior; (2) the relation between the trait and the answers that a person gives to test items can be described with an increasing monotonous function called the *Item Characteristic Curve* (ICC).

The very first models were known as Normal Models, because the ICC was described by the normal distribution function (Lord, 1968). Logistic models were introduced later to avoid the mathematical complexity of evaluating the normal distribution. The most important logistic models are the one-parameter model ((Rasch, 1960), and the two-parameter and three-parameter models (Birnbaum, 1968). All of them are based on the independence local assumption, which states that if the aptitude that explains the test performance remains constant, the examinee's responses to any pair of items are statistically independent.

Birnbaum's three-parameter model states that the probability of a correct answer to a question Q_i, given a value of the knowledge level, is defined by

$$P_i(\theta) = P(U_i = 1 \mid \theta) = c_i + (1 - c_i)\frac{1}{1 + e^{-1.7a_i(\theta - b_i)}},$$

where a_i, is called the discrimination index, b_i is called the difficulty degree, c_i is called the guessing factor[2], and U_i is a binary random variable that takes the value 1 if the student correctly answers question Q_i and 0 otherwise. Notice that: i) P(-)= c_i; ii) P''(b_i)= 0, i.e., the point of

[2] In the interactive tutorial in (Rudner 1998) it is possible to play with these parameters to obtain a better understanding of their meaning.

inflection of P corresponds to the difficulty degree; iii) a_i measures the relative "flatness" of P(θ). As an example, this function is plotted in Figure 2 with a_i=1.2, b_i=5, and c_i=0.25.

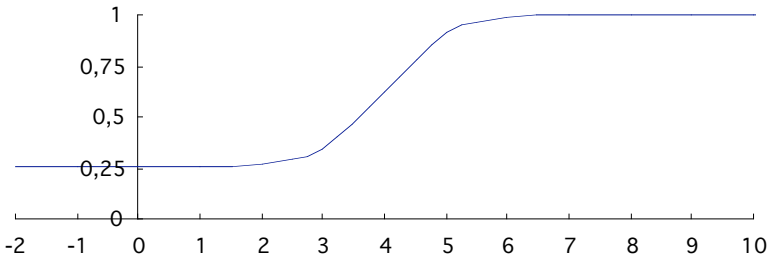

Fig. 2. ICC Graphic

This response model will be used to obtain the estimation of the trait θ. As explained in (Wainer & Mislevy, 1990), there are several methods to do this:

- *Maximum likelihood method* (Lord, 1980), which consists of finding the value of θ that maximizes the likelihood function;

$$L(\mathbf{u}|\theta) = L(u_1,...,u_n|\theta) = \prod_{i=1}^{n} P_i(\theta)^{u_i} (1 - P_i(\theta))^{(1-u_i)},$$

  where $\mathbf{u} = (u_1, ..., u_n)$ is the vector of the examinee's answers, that is, for i = 1,...,n, u_i is 1 if the answer to the i^{th} item is right and 0 otherwise, and $P_i(\theta)$ is the probability of correctly answering item i when the proficiency level is θ.

- *Bayesian methods*, which compute the ability level for which the posterior distribution is maximum. This posterior distribution is proportional to the product of the likelihood function and the a priori density function, that is[3]:

$$P(\theta/u) \quad L(\theta/u)f(\theta).$$

With respect to item selection methods, a good discussion can be found in (Van der Linden, 1998). We will only mention the most commonly used, which are:

- The *Maximum Information Method* (Weiss, 1982). This consists of selecting the item that maximizes the item information for the provisional proficiency level estimated until that moment, where the *Information Function* for the i^{th} item given the current estimation of the proficiency level $\hat{\theta}$ is given by:

[3] Usually, Bayesian methods assume the normal distribution for the ability level θ, so f(θ) is the normal density

$$I_i(\hat{\theta}) = \frac{[P_i'(\hat{\theta})]^2}{P_i(\hat{\theta})[1 - P_i(\hat{\theta})]},$$

- *Bayesian methods*. These are like the one proposed by Owen in (Owen, 1975), which selects the question that minimizes the posterior expected variance of the ability distribution.
- *Methods based on the difficulty level*. These are like the one proposed by Owen in (Owen, 1975), which selects the question whose difficulty level is closer to the current estimation of the student's knowledge level. Owen also showed that, for a continuous logistic ICC, the Bayesian method is equivalent to the difficulty-based method.

Having presented the theoretical background, we now describe the SIETTE system in detail.

THE SIETTE SYSTEM

The SIETTE system (http://www.lcc.uma.es/siette) is one of the components of the TREE Project (**TR**aining of **E**uropean **E**nvironmental Trainers and Technicians), funded by the EU Leonardo da Vinci Program, whose 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: an *Expert System* (ES) to classify and identify the species, the *Intelligent Tutoring System* (ITS) to assist students in the process of learning the domain, and the *Test Generation System* (TGS), to help them to know if they have reached the adequate level of knowledge. All these tools make use of the *Knowledge Base* (KB) which contains the information about the botanical domain and is being incrementally built using web forms by different users in diverse locations. Each component, including the KB, has an independent web-based interface that allows the whole system to be used both as a learning tool and as an independent consultation tool. We can see in Figure 3 the basic architecture of the TREE project.

SIETTE can be used in two different ways:

- Teachers and/or domain experts can use SIETTE to develop the tests, that is, to define their topics, questions, parameters, and specifications.
- Students can use SIETTE to take the tests that are automatically generated according to the specifications provided by the test designer. As the student answers the items the new ability level is computed and the next question selected, until the stopping criterion is met[4].

[4] The version of SIETTE described in this paper can be tried out at http://capella.lcc.uma.es/siette. The TREE system that integrates SIETTE is located at http://www.lcc.uma.es/TREE. A new version of SIETTE is available at http://www.lcc.uma.es/siette. In order to take a sample test, you only need to register as a new user (just type an identifier and a password and the system will create a student profile for you). For a demo of the test editor, send an e-mail to conejo@lcc.uma.es.

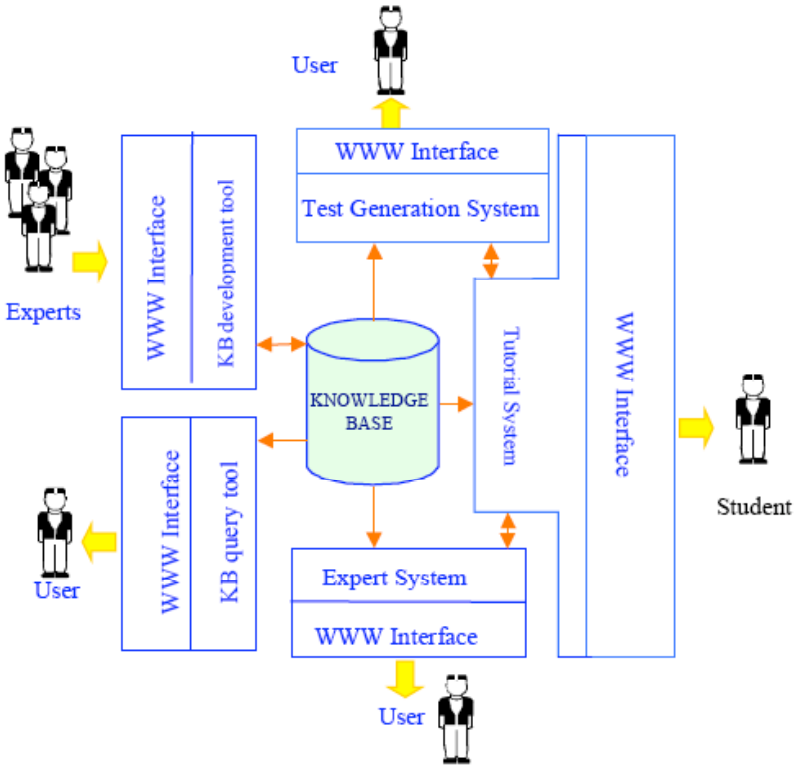

Fig. 3. Architecture of TREE

The components of the SIETTE system are:

- The *question knowledge base,* which is a database of questions or items related to a test. All these questions are calibrated with some parameters, which will be described in detail in the next section.
- The *test edition module,* which is the tool used by instructors or domain experts to define the tests and the structure of the subject domain: topics, questions, relationships between them, and relative weights of topics in the test. This information is stored in the question knowledge base. In this module the test developer can also define *test specifications,* that will guide the item selection process, and the finalization criterion (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, the *Validation and Activation* module checks the test specifications and characteristics to ensure that it is correctly defined (for example, it is not possible to define a test in which the minimum number of questions to be posed is 20 with a knowledge base of only 15 questions). Then the test is activated, that is, it

is made available to students that need to take it. These validation and activation processes are performed offline in the server side.

- A *temporary student model,* which is created and updated by SIETTE for each student that takes the test. Basically, the student model consists of a vector of K probabilities (p_0, p_1, ..., p_{K-1}), where p_i represents the probability that the student has reached a certain knowledge level i about the domain, where 0 is the lowest level and K-1 is the expert level. In addition, the student model stores information about which questions have been asked by the system.
- The *test generator* is the main module of SIETTE. It is responsible for selecting the questions that will be posed to each student. The generation process will be guided by the temporary student model and the test specifications.

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. The architecture of SIETTE is depicted in Figure 4.

Once we have presented the general structure of the SIETTE system, we will describe the test editor, the temporary student model, and the test generator in more detail.

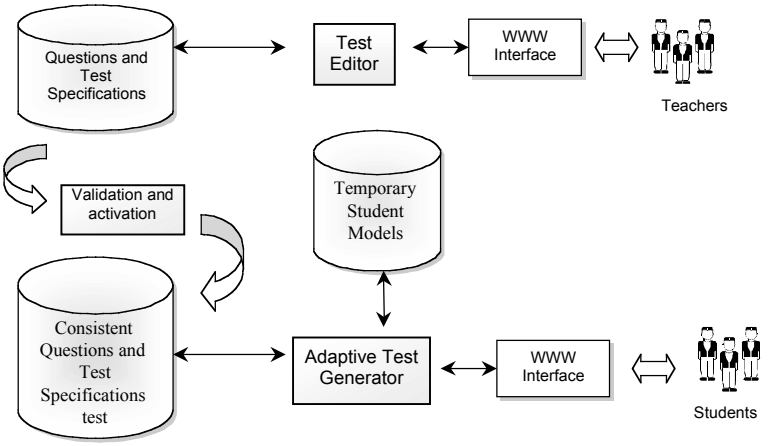

Fig. 4. SIETTE Architecture

Test editor

The test editor is the tool used by instructors in the SIETTE system to design the test. The information is supplied by test designers by means of HTML forms, and then it is saved in a relational database so the test generator module can use it.

In SIETTE, tests have a curriculum-based structure. Each test for a particular subject is structured in *topics* and *questions*. As shown in Figure 5, a question can belong to different topics, and a topic to different tests.

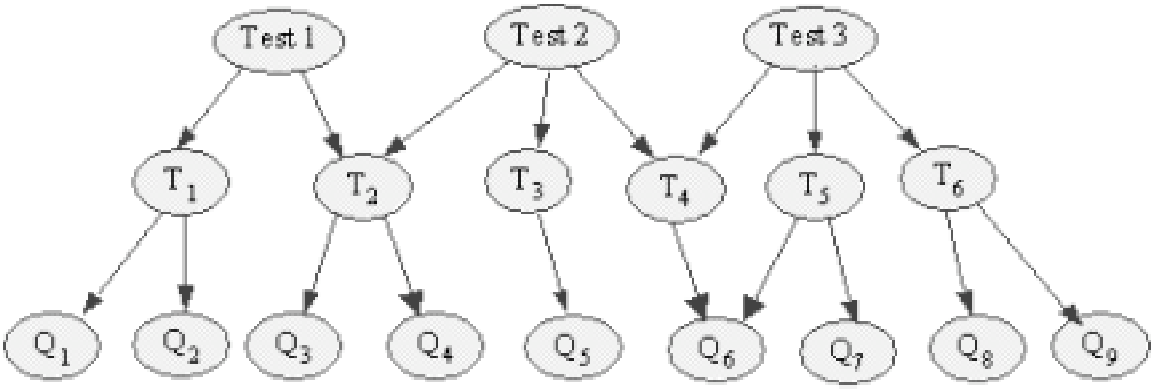

Fig. 5. Structure of tests

The curriculum defined in this way will be used by the item selection algorithm to generate content-balanced tests adjusted to the specifications previously defined, as described in (Kingsbury & Zara, 1989), (Welch & Frick, 1993), and (Huang, 1996). However, in the current version of the system this structure is not used in the proficiency estimation, that is, when the test finishes, the system provides an estimation for the knowledge level reached in the subject, but it does not have the capability to provide an estimation for the knowledge level reached in each of the topics of the test. As a more detailed information is sometimes required (for example, if this information is going to be used by an ITS to take an instructional decision), currently we are exploring the possibility of using or Bayesian Networks (Millán, Pérez-de-la-Cruz & Suárez, 2000), (Millán & Pérez-de-la-Cruz, 2002) or multidimensional IRT.

In Figure 6, we can see a snapshot of the interface for the test editor. In this test definition interface we can see that there are different interfaces to introduce tests, topics, and questions. Each node in the curriculum has some associated parameters that must be configured to complete the test design. We will review them in more detail:

Tests

In Figure 7, we can see the test definition interface.

The data required for a test are:

- *General test data.* These parameters are: the title, the date in which the test will remain active (if it is not provided, the test is always available), and an optional brief description.

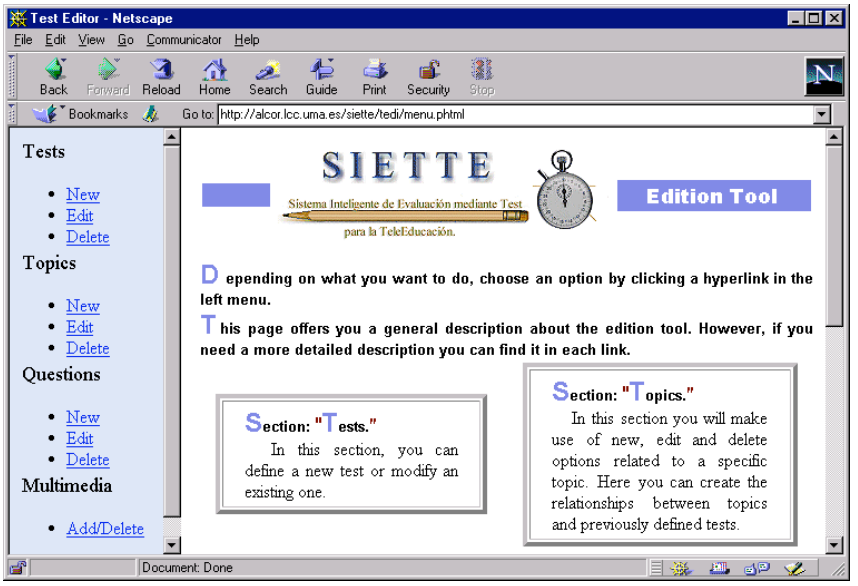

Fig. 6. Test editor interface

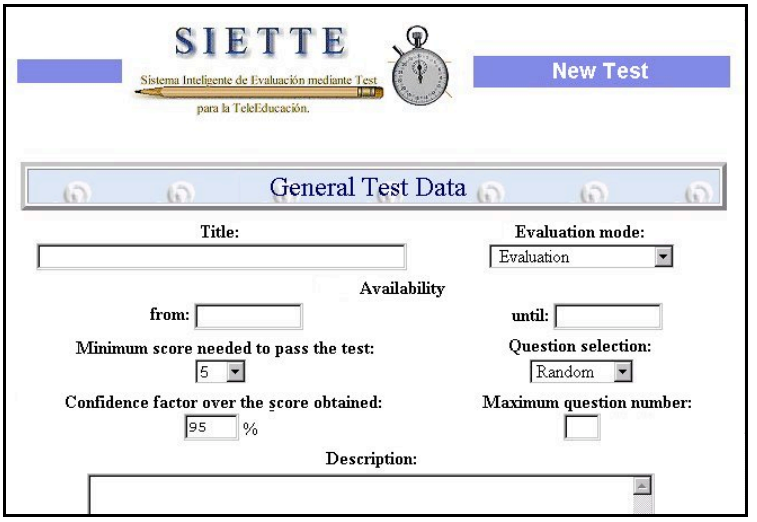

Fig. 7. Test definition interface

- *Assessment mode*. The system offers two different modes: *mastery* and *knowledge-level*. The *mastery* mode is binary in the sense that the system just evaluates whether the student passes the test or not, i.e., if he/she reaches the minimum score fixed by the instructor to pass the test. In the *knowledge-level* mode, the system determines the student's knowledge level with the confidence factor previously fixed by the

instructor, for example, the system determines that the knowledge level is 6 with a confidence factor of 0.9.

- *Item selection method*. The methods available in SIETTE are *Random* (questions are selected randomly), *Difficulty-based*, and *Bayesian.*
- *Maximum number of questions* to be posed in the test, which is optional.
- *Number of knowledge levels K.* The student's knowledge will be measured in terms of a discrete random variable with K values 0, ..., K-1. The value 0 will represent *no-knowledge* and the value K-1 will represent *perfect-knowledge*.

Topics

In order to design the test structure, the instructor will select the topics that will constitute the test. For each topic, two parameters are required: a *weight* which represents how important the topic is in the test, and *the minimum number of questions* of this topic that should be posed in a test (in random mode). The weighted relationship between tests and topics can be defined either in the test definition or in the topic definition interfaces (note that the same topic can belong to different tests). The interface for topic definition is shown in Figure 8.

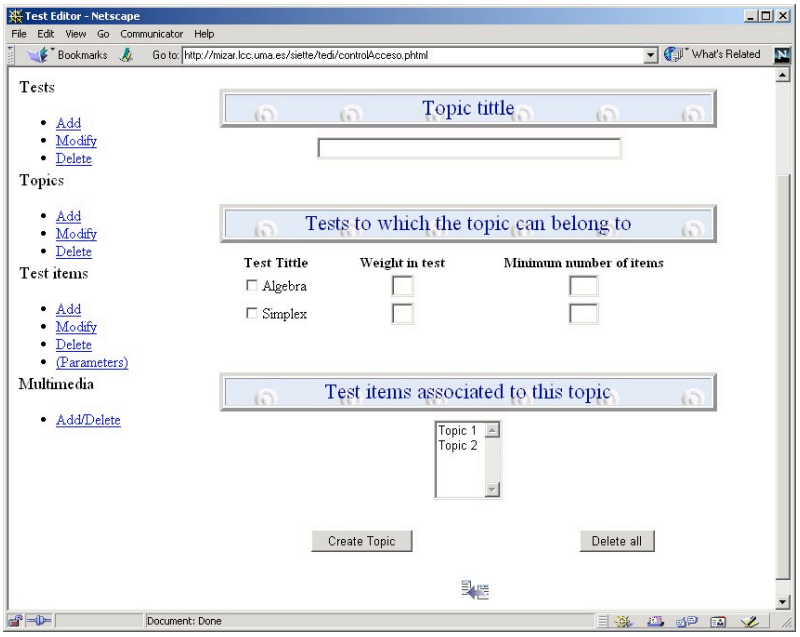

Fig. 8. Topic definition interface

Questions

Associated to each question, we have the following parameters:

- *General question data.* These are the title and the number of possible answers.
- *Question stem.* The question stem should be as simple as possible, and never confusing or ambiguous.
- *ICC parameters.* As we have already explained, it will be considered that the student's knowledge can be measured in terms of a discrete random variable θ that takes integer values in [0, *K-1*] instead of the continuous random variable used in classical IRT. This means that an ICC in SIETTE is a set $\{p_i\}$ of K values, that give the probability of a correct answer given that the student's knowledge is i, i.e., $p_i = P(U_i=1/\theta=i)$ for i=0,...,*K-1*. The discrete distribution is derived from an underlying continuous three-parameter logistic distribution, so a, b, and c values are also needed. The guessing factor c is determined automatically by SIETTE according to the number n of different answers that are shown to the student ($c = 1/n$). On the other hand, the instructor must define the discrimination index a and the difficulty degree b. The former must be a number between 0.5 and 1.5. Concerning the latter, since too easy or too difficult questions are hardly useful, only natural numbers between 0 and *K-1* are allowed.
- Extreme values of the ICC are not very significant, so only the "central" fragment has been discretized. Namely, and following some empirical adjustments, the interval [*(K-1)/2, (K+1)/2*] has been selected for discretization. The interval has been divided in *K-1* subintervals of equal length and the separation points give the values of the continuous variable θ that are in correspondence with the discrete variable; that is to say, $p_i = ICC(i - (K-1)/2)$ for i= 0, 1, ..., K-1.
- *Hint* associated with the item. This help will be presented whenever the instructor considers it is appropriate to give some orientation to the student.
- *Possible answers.* The instructor defines the correct answer and as many distractors as he/she wants, and also tells the system how many of these distractors should be shown in each question. Both the correct answer and the distractors can have a brief feedback. It is also possible to choose between some different presentations of the answers (in columns, tables, etc.) and to get a preview of the question in the web page before adding it to the database.
- *Topics* that the question is related to. This information can be supplied either in the topic or in the question definition interfaces.

Part of the question definition interface is shown in Figure 9.

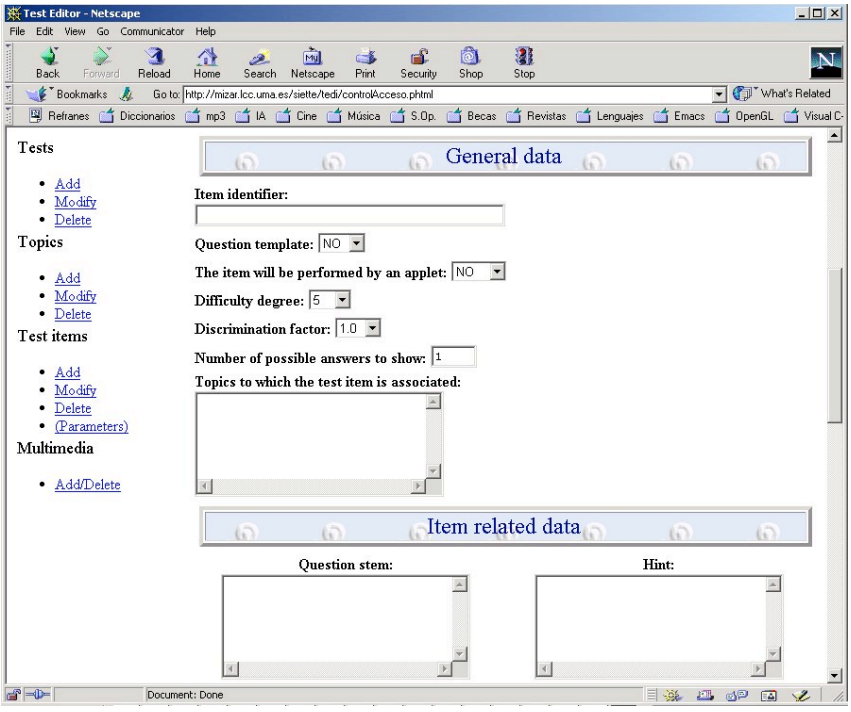

Fig. 9. Question definition interface

The SIETTE system also allows the definition of a *question template* by using HTML code or PHP. In this way, stems or answers containing variables can be defined. By randomly selecting values for these variables, the stem/answer will look different each time it is selected. So, for example, we can define a question template with stem "Which is the species shown in the following photograph?" and the photograph will be chosen randomly from a photograph database. In this way, many different questions can be defined with a minimum effort, improving the diversity and therefore the quality of the database.

Another interesting possibility is to use JAVA applets in order to include programs in the HTML pages to present the questions. This possibility has been extensively used, for example, in the application of SIETTE for cognitive abilities estimation (Arroyo, Conejo, Guzmám & Woolf, 2001).

There are two different ways to include applets in a question:

- The first possibility is to incorporate the applet in the stem/answer section. In this way, it is possible to define questions in which the stem (or possible answers) includes an applet that shows something (such as a simulation of a physical phenomena), and the student is asked to select the correct option after having observed the applet. This mechanism allows SIETTE to measure abilities that would be difficult to measure with traditional PPTs like sensorial, visual, or auditory capabilities or even perception and attention. These types of questions can be defined and used in a SIETTE test by simply coding the applet and including it (with the editor) in the section corresponding to the stem or the answers.

- The second possibility consists of letting the applet perform the assessment, that is, it is the applet itself which determines whether or not the answer is correct. In this case, the system will pose a question that contains a small program that is executed and shown to the student, who gives his/her answer interacting with the applet. Then the applet determines the correctness of the answer, and passes this information to the SIETTE inference mechanism, as illustrated in Figure 10.

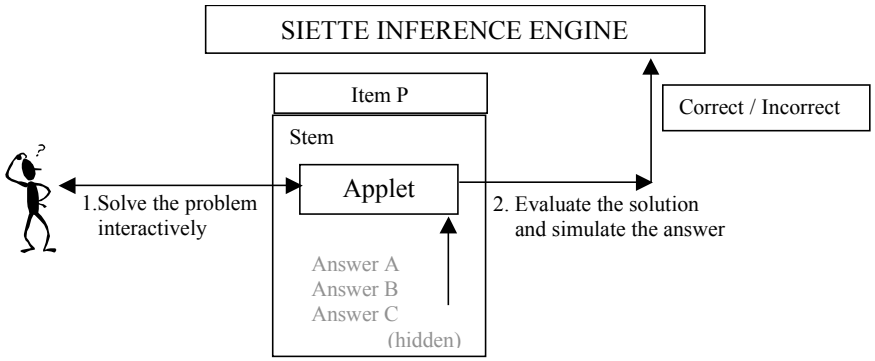

Fig. 10. Evaluation of an answer using applets

These types of questions can be combined with multiple-choice ones in the same test, and by using them new possibilities are opened up: it is possible to minimize the effect of guessing factors, or to control valuable information like answering time, or to measure knowledge that was difficult to evaluate using a multiple-choice format. Figure 11 shows an example of these types of questions (in a Botany domain). The goal of this question is to determine whether the student knows the geographical distribution of certain vegetable species. Using a template, a species is randomly chosen. Once this species has been selected, a map of Europe is shown, and the student must colour green (dark grey in 0) its corresponding geographical zone (using a paintbrush like in drawing software). Once this task is finished, the applet determines if the answer is right or wrong (within the margin of error allowed), and informs the system about the result of this assessment. The question in this screen has already been answered and evaluated, so the labels that appear in the bottom part of the screen (15.5% and 91.66%) correspond to the proportion of correctly located area.

One of the options that the SIETTE system offers to the user is the possibility of checking the answer to each item right after having answered it. In items controlled by applets the student can use the *Correct* button in order to see the correct solution.

From the test designer's point of view, defining these types of items is very easy. The applet programmer only has to write two functions: a function called *Solve* that will show the correct solution to the student, and another function called *Evaluation* that evaluates if the answer is correct. For example, in the previous example, the *Evaluation* function would compute the proportion of the area that has been correctly classified, and will consider the answer correct if the proportion is over a certain threshold previously determined by the test designer.

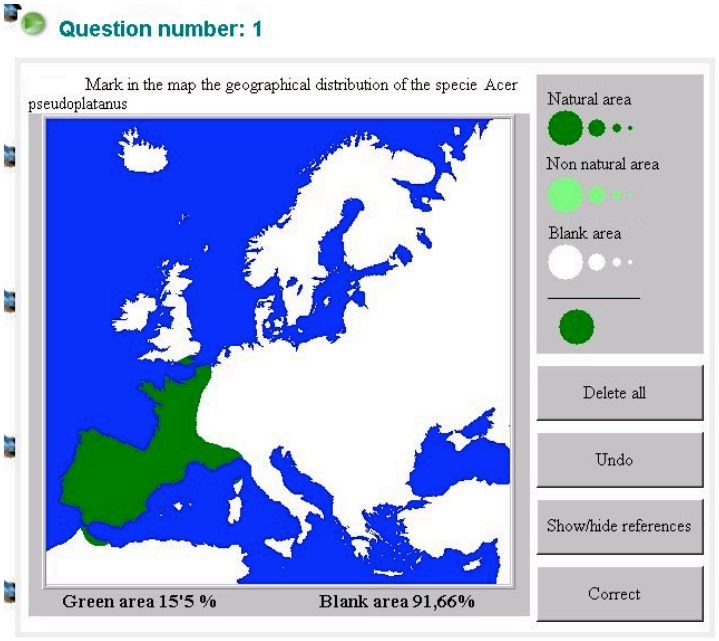

Fig. 11. An example of a question whose answer is evaluated by an applet

To sum up, 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.
- The possibility to use *Multimedia content* in questions and possible answers.
- By using question and answer *templates*, a great number of different questions (or answers) can be automatically generated by the system.
- By using *JAVA applets*, the student can answer questions that would be difficult to use in traditional PPTs.

The 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.

To present an example of a temporary student model, we will assume again that the number of knowledge levels fixed by the test designer is K=11. Then, the student's knowledge level would vary from Level 0 (*novice*) to Level 10 (*expert*). Initially, and in the absence of any other information, the probability is uniformly distributed among the 11 levels and, as the student takes the test, they are updated with a Bayesian procedure. In Figure 12, we can see an example of a temporary student model and its structure.

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}	$A_{20,1}$

Fig. 12. An example of a temporary student model

We can see that the system keeps track of different types of data regarding this particular student:

- Date that the test was taken, knowledge level reached, and confidence interval associated.
- Probability that the student has reached each knowledge level.
- Topic distribution in the test posed.
- Questions that have been selected (these questions are selected according to the bayesian procedure) and presented to the student.

Once the test is finished, the temporary student model becomes the student model, and all the information it contains is stored until the next session.

Test generator

The test generation algorithm is similar to the one presented in 0 and consists basically of three main procedures: (a) item selection, (b) knowledge level estimation, and (c) termination criterion. This module has been implemented using a CGI application.

Item selection

Test developers can choose between three different item selection procedures: a) *Bayesian procedure* (which selects the item that minimizes the posterior standard deviation), b) *Difficulty-based procedure* (which selects the item that gives the minimum distance between the mean of

the ICC and the mean of the current student's knowledge distribution)[5], and *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 criterion does not allow differentiating between two questions, the question is selected randomly between the possible alternatives. This happens usually when using templates, because every instance of a template has the same ICC.
- *Content balancing*. To assure content balanced tests, SIETTE uses the weights specified by the test designer for each topic. 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*. The 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.

Knowledge Level Estimation

Once the best question has been chosen, the system poses it to the student. Next we will describe how the SIETTE system computes the student's new proficiency level once he/she has answered the question. To compute the posterior distribution, SIETTE applies Bayes' Theorem, using the K values given by the ICC $\{P(U_i=1/\theta=k)$, for $k = 0,...,K-1\}$ and the a priori knowledge distribution $\{P(\theta=k), k=0, ..., K-1\}$ (which is set to be uniform, i.e., $P(\theta=k)=1/K$ for each k=0, ..., K-1). Then:

$$P(\theta=k/U_i) = \frac{P(U_i/\theta = k)P(\theta = k)}{\sum_{i=0}^{K-1} P(U_i/\theta = i)P(\theta = i)}, \qquad \text{for each } k=0, ..., K-1.$$

Termination Criterion

The termination criterion is also determined by the test developer, and it can be any valid combination (using OR) of the following cases:

1. The standard deviation of the distribution of the student's knowledge is smaller than a fixed value (the estimation is accurate enough).
2. The probability that the student knowledge is greater (smaller) or equal than a fixed proficiency level is greater than a fixed number. The idea is that the test finishes when the proficiency level is bigger (smaller) than a certain level determined by the test designer. To

[5] As already mentioned in the section devoted to CATs, for a continuous logistic ICC the Bayesian and difficulty-based criteria are equivalent (Owen, 1975). In the section describing the evaluation of the system, we will show that in our case, both criterions give similar results.

give an accurate assessment, the instructor determines the desired precision of the measure by specifying a confidence factor. This termination criterion is used if the mode chosen is "mastery".

3. The system has already posed all the questions in a test and/or
4. The system has posed at least the minimum number of questions of each topic specified by the test designer.

One of the characteristics of the system is the capability of providing immediate feedback. While a student is taking the test, the system can show him/her the solution to the question he/she has just answered. At this moment, the student could try to cheat the system by pressing the navigator BACK button. To avoid this behaviour, we have implemented a simple state machine that is shown in Figure 13.

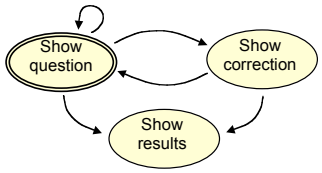

Fig. 13. State machine for avoiding cheating

AN EXAMPLE

In this section, we will present an example (a TREE test about a botany domain) to illustrate the functioning of the system when a student takes a test. Again, we will consider that the number of knowledge levels is K=11. In the test specifications for this particular example, the minimum level of knowledge required in order to pass the test has been set to 5[6], and the level of confidence to 75%.

Initialization of the temporary student model

Initially, and in absence of any type of information, the knowledge level of the student is considered to be a uniform distribution, that is, the probability that the knowledge level of the student is i (for i = 0, ...,10) is 1/11, as shown in Figure 14.

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 expected variance. In Figure 14, we can see the first question selected, and its associated ICC. As the

[6] Traditionally, the Spanish evaluation system considers a minimum of 5 points over a total of 10 points in order to pass an exam.

initial level is set to be uniform, the student has equal probability of belonging to each of the eleven levels (1/11).

Selection, presentation, and evaluation of successive questions

Now, the student answers the question, and the system updates the temporary student model and uses it to select the next question. In Figure 15, we show an intermediate state after the student has correctly answered seven questions. We can see that, as expected, higher knowledge levels are now more probable.

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 16: the student's knowledge level is estimated to be level 9 with probability 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.

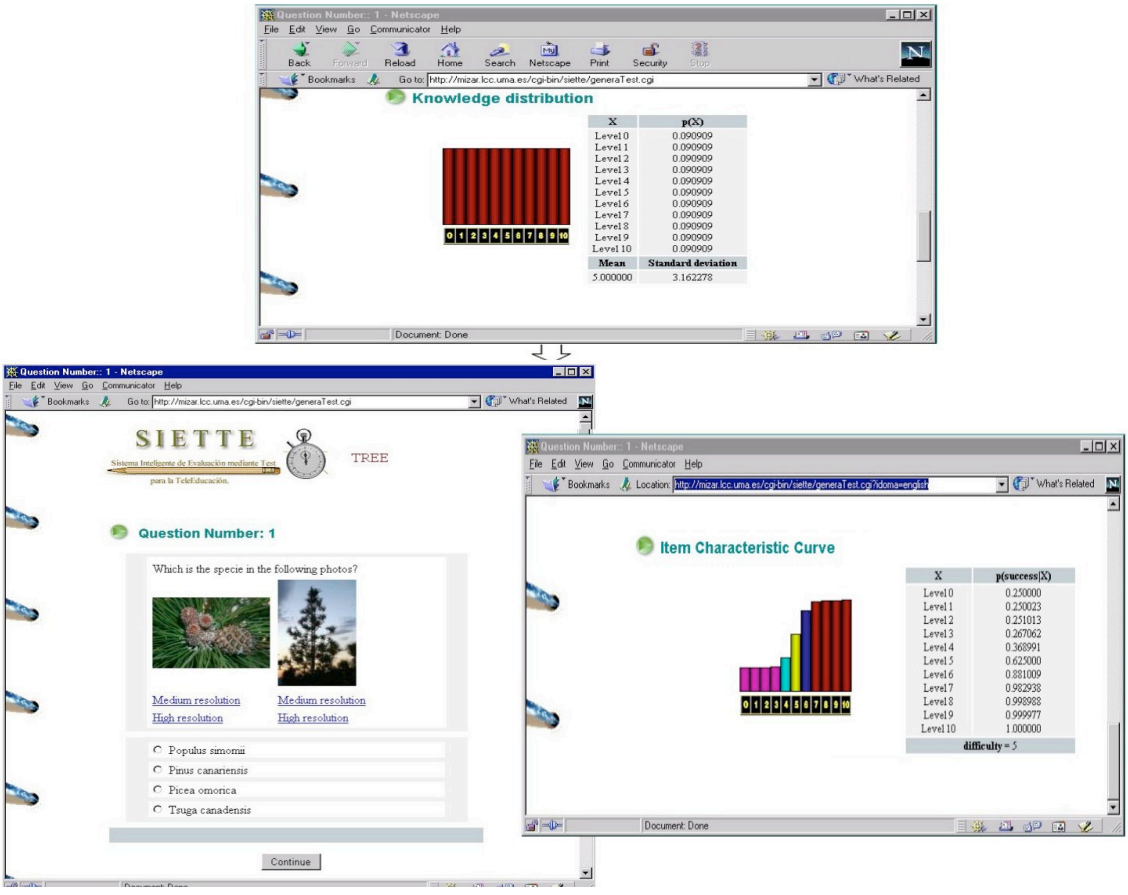

Fig. 14. Initial state in a test session, first question presented, and its associated ICC

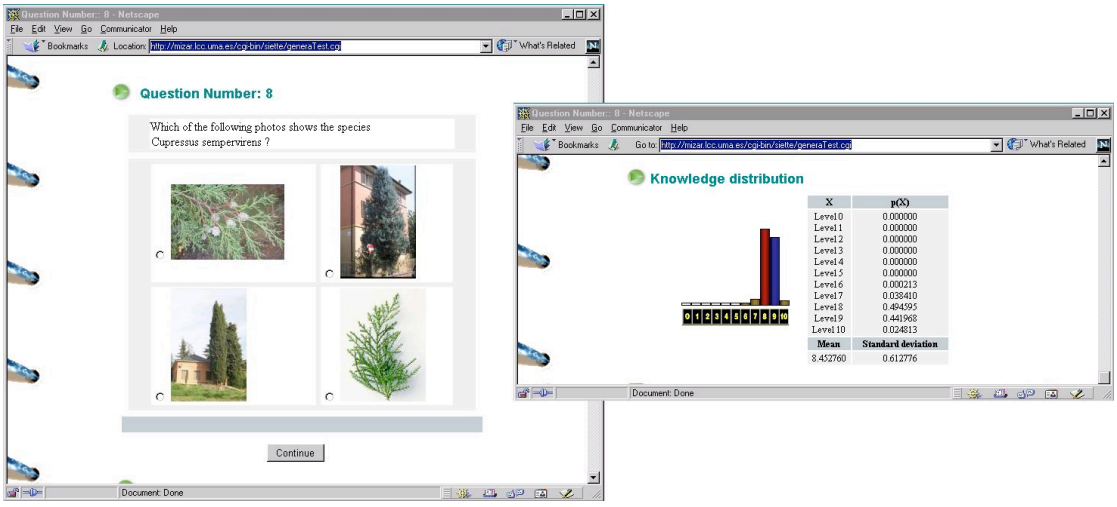

Fig. 15. Question 8 and knowledge distribution after seven questions

EVALUATION

In order to test the validity of the approach presented and the usefulness of the system, two different studies were performed:

1. An empirical study using simulated students, whose goal was to test the validity of the ITR discretizations implemented in the SIETTE system.
2. An informal evaluation study with real students, whose goal was to test the usefulness and quality of the SIETTE system itself.

Next we will describe in more detail the conditions of each study and present the results.

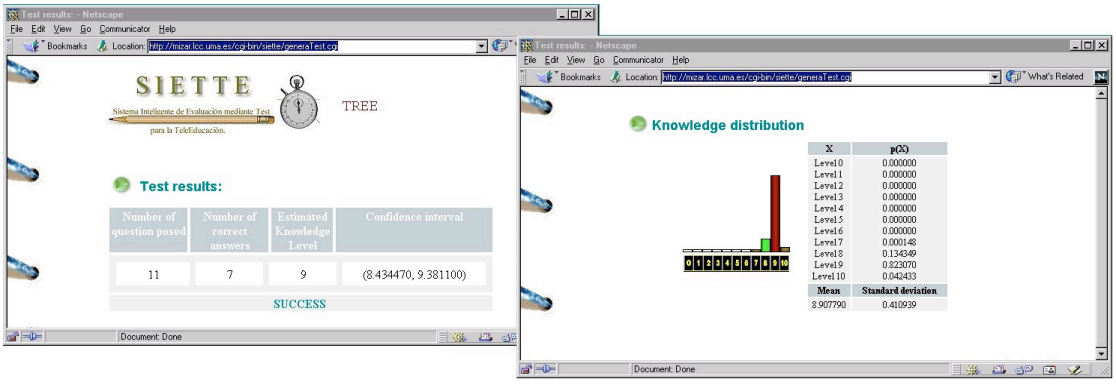

Fig. 16. Final test session view

Evaluation with simulated students

In this section, we will describe how our approach to adaptive testing has been evaluated using simulated students, as proposed in (VanLehn, Ohlsson & Nason, 1995). As we have already explained, SIETTE implements the IRT model assuming that the student's knowledge θ can be represented as a random variable that takes integer values in [0,K]. The goal of this study was to evaluate the performance of this discretization of IRT.

Each student is represented by his/her θ value. The simulation begins with the random generation of a population of N students, i.e., with the generation of N values for θ. These values are considered constant during the test (i.e., no student learning occurs while taking the test). In the simulations described here the population has been generated to be uniformly distributed in 0, ..., K[7].

The simulator uses a set of Q void items and their associated ICCs. Each ICC is given by a set of K+1 values that represent the conditional probabilities of giving the correct answer to the question, given that the student belongs to each of the K+1 classes, and are computed as previously explained (see section describing item parameters). The ICC parameters can be estimated or generated randomly. The item pool is assumed to be correctly calibrated.

Initially, the knowledge distribution is uniform, i.e., the student has the same probability of belonging to each of the K+1 classes. The posterior probability is computed applying Bayes' rule. In the simulations, the test finishes (in the general case) when the probability of belonging to certain class reaches a fixed threshold γ (close to 1), meaning that the student belongs to the class with a confidence factor greater than γ.

The student's behaviour during the test is determined according to his/her current estimated knowledge level and to the conditional probability that a student of this knowledge level solves the question correctly. These two values are used to compute the probability p that the student will correctly answer the proposed item. Then, a pseudo-random uniformly distributed value q in [0,1] is generated. If q>p, the system will consider that the student correctly answered the question. Once the test has finished, it is considered that the student was correctly classified if the predicted knowledge level is equal to the real knowledge level.

Three different experiments were run. The goal of these experiments was to study the influence of different test settings in the test accuracy (measured in terms of the percentage of correctly classified students (A) and test length (T). The test settings that were evaluated in each experiment were:

- *Experiment 1.* Number of classes (K) and confidence factor (r)
- *Experiment 2.* ICC parameters (a and c)
- *Experiment 3.* Item selection criteria (adaptive versus random)

Next we will describe in detail the conditions of each experiment, and present and interpret the results.

---

[7] Other distributions were also tried, not yielding significant differences in the outputs.

Experiment 1

In this experiment a=1.2, c=0, and b was randomly generated, so there were the same number of questions of each difficulty level. As already explained, the goal of the experiment was to study the influence of K and in test accuracy and length. Questions were selected randomly. The results are shown in Table 1.

Table 1
Results of the first experiment

	Confidence factor = 0.75		*Confidence factor = 0.90*		*Confidence factor = 0.99*	
Number of classes K	*% of correctly classified students*	*Average number of questions posed T*	*% of correctly classified students*	*Average number of questions posed T*	*% of correctly classified students*	*Average number of questions posed T*
3	84.05	2.00	95.82	3.58	99.46	5.65
5	81.61	6.23	92.76	10.38	99.37	19.27
7	80.96	11.11	92.85	18.16	99.38	33.12
9	80.86	16.15	92.93	26.39	99.42	47.27
11	80.52	21.19	92.92	34.54	99.26	60.85

The interpretation is that, even with a correctly calibrated item pool, it is not easy to classify "all" the students correctly. This is due to the model itself, that assumes that it is possible (but with a low probability) that a student with a high knowledge level will answer an easy question incorrectly. The results also show that the percentage of correctly classified students depends more on the confidence factor required than on the number of classes used. On the other hand, the number of questions posed is strongly related to the number of classes considered. For practical reasons, the test should have as few questions as possible, because long tests would be too boring for real students. This practical consideration leads to a compromise between the number of questions and the number of classes.

Experiment 2

The goal of the second experiment was to study the influence of the parameters a and c and test length and accuracy. Tables 2 and 3 show the results (=0.9 and K=7).

Table 2
Guessing factor influence

Guessing factor c	*% of correctly classified students*	*Average number of questions posed T*
0.00	92.85	18.16
0.10	92.37	25.34
0.25	92.11	36.05
0.33	91.73	43.37
0.50	91.49	63.37

Table 3
Discrimination index influence

Discrimination index a	*% of correctly classified students*	*Average number of questions posed*
0.20	90.4	174.9
0.50	91.5	35.2
0.70	91.9	26.3
1.20	92.8	18.1
1.70	93.8	15.3
2.20	95.4	14.8

As expected, a small guessing factor and a big discrimination index give the best results. These results also show the great influence of the guessing factor c in the number of questions needed. The discrimination index, a, does not have such a great influence in the number of questions if it is bigger than certain threshold. For values smaller than that threshold, the number of questions needed grows very fast. That means that items with a low discrimination index are not informative enough and therefore yield too long tests.

Experiment 3

The goal of this experiment was to study the influence of the item selection criteria. Table 4 shows the empirical result obtained with the simulator (θ=0.9).

Table 4
Accuracy of the CAT approximation

	Bayesian		*Difficulty-based*		*Random*	
Number of classes K	*% of correctly classified students*	*Average number of questions posed T*	*% of correctly classified students*	*Average number of questions posed T*	*% of correctly classified students*	*Average number of questions posed T*
3	96.06	3.58	95.62	3.58	95.82	3.58
5	93.31	6.87	94.67	7.37	92.76	10.38
7	92.75	8.70	94.43	9.03	92.85	18.16
9	92.53	9.85	94.23	10.14	92.93	26.39
11	92.10	10.71	94.14	11.02	92.92	34.54

As we can see, adaptive item selection criteria (Bayesian and difficulty-based) greatly improve the performance, as the number of questions needed is always smaller than that corresponding to the random criterion. The greater the number of classes K, the better the performance of adaptive criteria versus the random criterion. These results encourage the use of a CAT procedure. The difficulty-based criterion has been chosen over the Bayesian one because it gives similar results, but its computational cost is much smaller. Similar results were obtained with other discrimination and guessing factors.

Evaluation with real students

In June 2000, an informal evaluation study was conducted in the Computer Science Department of the University of Málaga in Spain. The students were taking the subject *Artificial Intelligence*, in which they learn the basics of Artificial Intelligence and the language LISP[8]. A test to assess LISP knowledge was developed and introduced in the SIETTE system by teachers of the subject. The students were at the end of the semester, so they had received about 60 hours of teaching in the subject and were to take their exam in approximately four weeks. Students were taken to the lab to take an on-line test about LISP and then filled a questionnaire about their experience. The

[8] The course is traditional and does not have any on-line component

results of this test did not have any influence in their final grade. Participation in the study was voluntary and anonymous. Twenty-four students participated.

The questionnaire for the study is provided in Appendix A. Only one student reported extensive experience with the subject, while approximately the same number of students declared themselves to have no experience other than the lectures (45.83%) or some personal work (50%). Most of them learned to use the system very quickly (87.5% needed less than five minutes and the rest between five and ten minutes). Regarding the usefulness of the system in the learning process, the majority considered it quite useful (50%) or very useful (12.5%), while only 4.17% and 8.33% ranked the usefulness as "not very much" and "so and so", respectively. The more commonly cited reasons for usefulness were: "it is practical", "it is complete", "it is a good complement to lectures", while reasons against usefulness were "the same item schema is repeated too often" (which seems to concern the LISP test used in the evaluation more than the usefulness of the system) , "the level of difficulty is different than that required in the exam"[9], "there is no feedback" or "the system does not teach" (as said before, the system allows the possibility to include feedback, a possibility that was not exploited in this particular test used in the evaluation. So, these last two complaints could be solved just by including the proper feedback in each question when defining the test).

When asked if they intended to keep on using SIETTE, 50% said yes, 37.5% said they did not know, and 12.5% said no. Most of the students who answered "I do not know" or "no" said that it was too difficult to get access to the Internet in the school and that they could not afford to use it at home, while students who said they were going to use the system declared that it helped them to check if they had learned the subject and that it was a good training for the exam. The great majority of the students said that they would recommend SIETTE to other students (87.5%), while only 4.17% said they did not know, and 8.33% said they would not. All students considered the interface either easy (66.67%) or very easy (33.33%) to use, and the more commonly cited reasons were its simplicity and good design. Only one student said that more information about statistics and graphics was needed.

When asked about what they liked most about SIETTE, students mentioned the intelligent generation of questions, the simplicity and design of the interface, the completeness of the system, the variety of questions, web accessibility, and the graphics showing their performance (i.e., the graphics showing the evolution of their knowledge distribution). Regarding possible improvements, they mentioned that the same question schema was used too often, that there were very few tests to take, and the lack of feedback. Some of them suggested that there should be a downloadable version of the system so they could use it in local mode, and also that there should be a way to download all the questions in the database (which in our opinion might help them to pass the exam, but would go against the philosophy and purpose of our system).

---

[9] This seemed to be a great concern for the students that participated in the evaluation. After the exam, a student that had not passed the test complained to his teacher about the higher level of difficulty of the exam. After some conversation with him, we found the reason: he explained that he had been using the system to try to see all the questions in the database. To this end, he did not answer any of the questions posed. However, as the test is adaptive, the system never showed him the more difficult questions, as his assumed level of knowledge was always low.

Only two students reported having found mistakes, but the mistakes were not relative to the functioning of the SIETTE system but to the LISP test (in some questions there were some bugs, either in the question stem or in the possible answers).

Regarding the grade obtained, 16.66% of the students considered it as unfair, while 41.67%, 20.83%, and 4.17% considered it right, fair or totally fair, respectively. The difficulty of the questions was considered normal by the majority of the students (58.33%), and difficult or easy for the rest (12.5% and 4.17%). Only two students compared the grades obtained and expected: a student who declared no previous experience with the subject other than the lectures said that he expected 7.5 (out of ten points) and that he got a 4.5, and a student who declared some personal work said that he expected 9 and obtained 7. It seems that their estimations of the grade obtained were rather optimistic in relation to the amount of work devoted to the subject.

Finally, there was a tie between SIETTE and a traditional paper and pencil test (41.67% of the students preferred each of the options). However, many of the students declared that they preferred a traditional test only until the SIETTE system was fully developed and tested. Not many additional comments were made; only one student reported that in his opinion the penalty for incorrect/not answered questions was too high, and some other students took the chance to complain about the difficulty of using the Internet due to the lack of free use laboratories in the Computer Science School[10].

On the other hand, we have not conducted any study with the other types of users of our system, that is, teachers or test developers. However, several teachers have used the system to develop their tests. Currently, there are tests available for several subjects, such as Piagetian Tests (Arroyo et al., 2001), Language Processors, Botany, LISP, etc. Test developers reported positively about the quality and design of the test editor interface that, in their opinion, made the definition and modification of tests very easy.

RELATED WORK

There are many systems, commercial and non-commercial, that have been developed to assist in the creation of tests. In addition, some learning environments and tutoring systems offer interesting test modules. In this section we will briefly review some of them, with special emphasis on whether they have adaptive capabilities or not.

In Intralearn (Intralearn, 2001), WebCT (WebCT, 2001), and Webassessor (Webassessor, 2001) the tests are static. Teachers decide which items should be included in each test and in what order. For example, in WebCT the items are stored in an item pool and grouped in categories. An exam can contain items from different categories, and each item can be part of different tests. On the contrary, in Intralearn each question defined by the teacher belongs to a unique test, although it can be associated to any subtopic in the curriculum. In Webassessor questions also belong to a unique test but the position of both the different items and their possible answers can be determined by the test designer or randomly. There are some other non-adaptive systems that allow a certain flexibility in the generation of tests. For example, WebCT,

[10] Fortunately this deficiency was solved in the Academic Course 2000-2001.

QuestionMark's Perception (QuestionMark, 2001), and TopClass (TopClass, 2001) have different item pools for each topic that are used in the quizzes, that can be generated by the test designer or randomly. In that way, each student will receive a different set of questions. In QuestionMark´s Perception, the test designer can create "jumps" to specific locations within the test based upon responses to individual questions or scores. This adaptive-branching feature provides the system with some degree of adaptativity.

The more sophisticated systems use an overlay student model to select the next question to ask. For example, in the Lisp Tutor ELM-ART (Weber & Specht, 1997) there is a testing component that uses information about the performance of the student in previous tests to select the next question to ask. Medtec (Eliot, Neiman & LaMar, 1997) is a web-based intelligent tutor for basic anatomy. Tests are generated automatically based on an overlay student model. In (Lee & Wang, 1997) a hypermedia learning system is presented. A fuzzy approach is used to diagnose the student's answers and to create and update an overlay student model. This student model is used to select the problem to be presented to the student. The selection of problems in these three systems is based on the same idea: select problems according to their difficulty and the student's performance (the better the performance, the more the difficulty). Even such a simple strategy can result in tests that are challenging for the students, but of course the use of IRT further exploits this possibility achieving much better results.

From the designer's point of view, the more commercial systems might be more versatile for the test design. Currently, SIETTE only admits *multiple-choice* questions whereas other systems offer more options: *true/false* and *short answer* in Intralearn, Webassessor, QuestionMark´s Perception, WebCT, TopClass, and ELM-ART; *fill blanks* in QuestionMark´s Perception and WebCT; and *calculated questions* in WebCT. However, as we have already discussed, the use of Java Applets opens up many possibilities, since virtually any type of question can be included in the test by means of Java Applets. The implications that the use of these new types of questions can have in students' assessment need further research.

However, though some of these systems offer tests that are highly customizable and allow the inclusion of many different types of questions, none of them implements IRT theory, and therefore the main advantages that SIETTE has over them are the classical advantages of CAT: more accurate diagnosis with reduced test length.

CONCLUSIONS AND FUTURE WORK

Since their advent, computers have been systematically used for educational purposes. The emergence and growth of Internet and the application of techniques such as hypertext or multimedia add new possibilities to traditional educational methods. In the field of assessment methods and, particularly, of Test Theory, CATs offer many advantages that have been widely studied in the related literature, namely, a decrease in the number of questions needed to evaluate a student (by adapting the test to his/her particular needs) and a more accurate estimation of his/her knowledge level.

The system described in this paper combines the dynamic nature of CATs with the advantages that the WWW offers as a learning environment and as a knowledge dissemination platform. By using the web, we have provided the Educational Community in the world with a tool to assist in the difficult task of assessment. The SIETTE system allows teachers to edit their

tests online, and assures a consistent and accurate assessment of their students. The possibility of including multimedia contents and applets allows us to pose questions that were difficult to manage and evaluate in traditional PPTs; for example, all subjects that involve recognition of objects from photographs or some sort of interaction. Another interesting feature of the SIETTE system is its capability of using question templates and also applets which can be shown in the stem and/or answers and even used to perform the evaluation of the question.

We would like to remark on the importance of using efficient algorithms to select the best question to ask and also to store results. Delay times due to these processes and to download times can keep students waiting for too long when using the system. In the evaluation of the SIETTE system with real students we could check that the delay times were affordable even with a great number of students connected simultaneously. To improve the average performance, it might be interesting to use the Internet time delay to run the algorithms to select the next question on the server side, while the student is still waiting or thinking about the last question on the client side.

The development of the SIETTE system is intended to allow the use of traditional CATs in web-based educational settings. This change of context has some potential problems that deserve a careful search for innovative solutions. Currently, our work is proceeding in the following directions:

- *Integration of CATs in ITSs.* To this end, it is necessary to extend SIETTE's capabilities so it can assess more than one variable at a time and therefore provide more detailed information, as needed in an ITS. This is a major problem with the SIETTE system, given that in its current form (based on unidimensional IRT) the only way to use it in an ITS is to develop specific tests for each part of the curriculum. Our work in this direction is based on the definition of *CATs based on Bayesian Networks* that allow diagnosing more than one ability at a time (Millán et al., 2000), (Millán, 2000), (Millán & Pérez-de-la-Cruz, 2002).

- *Item calibration.* Obviously, the quality of the tests will strongly depend on the accuracy of the estimations of the test parameters. Traditionally, these parameters are estimated by teachers or obtained from previous studies where many students are involved. The problem is that both procedures present their drawbacks: teacher's estimations are always subjective and therefore subject to error, and these types of studies are usually unaffordable in most educational settings. However, the fact that our system is available through the Internet opens up a new possibility: data concerning students' answers can be stored and used in the application of learning techniques to calibrate the parameters. In this way, the parameters can be learned automatically by the system (and therefore the quality of the test can be improved) as students take the test. The result will be a test that learns from students (in the same way a human teacher does) whose questions are more suitable for each student depending on his/her knowledge level. In this direction, we are using simulated students to learn the parameters. Using this simulator we have obtained encouraging preliminary results that suggest that further work could significantly improve the quality of the test item pool (Conejo, Millán, Pérez-de-la-Cruz & Trella, 2000) and (Ríos, Millán, Trella, Pérez-de-la-Cruz & Conejo, 1999).

- *Using a richer variety of item types*, like polytomous[11], true/false, fill-blanks, relate-concepts, past-time items, etc. Some of these items are automatically generated by the system using item templates (Belmonte, Guzmán, Mandow, Millán, & Pérez-de-la-Cruz, 2002).

We are also working in some technical improvements, like:

- *Different item editors* (stand alone, web-based, customized editors for each different subject).
- *Tools for statistical analysis of results* (grouped by items, students, tests, etc.)
- *Using XML representations* for database independence and data exchange, accomplishing current test standards.
- *Using test-lets* (i.e., selecting and presenting a group of questions in the same interface) to speed up the performance.

References

Arroyo, I., Conejo, R., Guzmám, E., & Woolf, B. P. (2001). An Adaptive Web-based Component for Cognitive Ability Estimation. In J. D. Moore, C. L. Redfield, & W. L. Johnson (Eds.) *Artificial Intelligence in Education. AI-ED in the Wired and Wireless Future* (pp. 456-466). Amsterdam: IOS Press.

Belmonte, M.V., Guzmán, E., Mandow, L.,Millán, E., & Pérez-de-la-Cruz, J. L. (2002). Automatic generation of problems in web-based tutors. In B. Howlet, B and L. Jain (Eds.) Internet-Based Teaching and Learning. Series on Advance Information Processing. Berlin Heidelberg: Springer-Verlag.

Birnbaum, A. (1968). Some Latent Trait Models and Their Use in Inferring an Examinee's Mental Ability. In F. M. Lord, & M. R. Novick (Eds.) *Statistical Theories of Mental Test Scores*. Reading, MA: Addison-Wesley.

Conejo, R., Millán, E., Pérez-de-la-Cruz, J. L., & Trella, M. (2000). An Empirical Approach to On-Line Learning in SIETTE. In *Lecture Notes in Computer Science 1839. Proceedings of 3rd International Conference on Intelligent Tutoring Systems ITS´2000* (pp. 604-614). Berlin-Heidelberg: Springer Verlag.

Eliot, C., Neiman, D., & LaMar, M. (1997). Medtec: A Web-based Intelligent Tutor for Basic Anatomy. *Proceedings of WebNet'97, Second World Conference of the WWW, Internet and Intranet* (pp. 161-165). ACCE.

Flaugher, R. (1990). Item Pools. In H. Wainer (Ed.) *Computerized Adaptive Testing: A Primer*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hambleton, R. K. (1989). Principles and Selected Applications of Item Response Theory. In R. L. Linn (Ed.) *Educational Measurement*. New York: MacMillan.

Huang, S. X. (1996). On Content-Balanced Adaptive Testing. In *Lecture Notes in Computer Science: Vol. 1108. Proceedings of 3rd International Conference CALISCE'96* (pp. 60-68). Berlin Heidelberg: Springer Verlag.

[11] Polytomous items are items which have more than one correct answer.

Intralearn. URL http://www.intranet.com [Accessed 2001, May 21].

Kingsbury, G., & Weiss, D. J. (1983). A Comparison of IRT-based Adaptive Mastery Testing and Sequential Mastery Testing Procedure. In D. J. Weiss (Ed.) *New Horizons in Testing: Latent Trait Test Theory and Computerized Adaptive Testing*. New York: Academic Press.

Kingsbury, G., & Zara, A. R. (1989). Procedures for Selecting Items for Computerized Adaptive Tests. *Applied Measurement in Education, 2*(4), 359-375.

Lee, S. H., & Wang, C. J. (1997). Intelligent Hypermedia Learning System on the Distributed Environment. In *Proceedings of ED-MEDIA/ED-TELECOM'97, World Conference on Educational Multimedia/Hypermedia and on Educational Telecommunications* (pp. 625-630). ACCE.

Lord, F. M. (1980). *Applications of item response theory to practical testing problems.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Lord, F. M. (1970). Some test theory for tailored testing. In W. H. Holtzman (Ed.) *Computer assisted instruction, testing and guidance* (pp. 139-183). New York: Harper and Row.

Lord, F. M. &. Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Millán, E. (2000). *Bayesian System for Student Modeling.* Doctoral dissertation, Universidad de Málaga, Dpto. de Lenguajes y Ciencias de la Computación. Available (in Spanish) at http://www.lcc.uma.es/~eva/investigacion/tesis.html

Millán, E., & Pérez-de-la-Cruz, J. L. (2002): A Bayesian Diagnostic Algorithm for Student Modeling and its Evaluation. *User Modeling and User Adapted Interaction*, 12, 281-330.

Millán, E., Pérez-de-la-Cruz, J. L., & Suárez, E. (2000). An Adaptive Bayesian Network for Multilevel Student Modelling. In *Lecture Notes in Computer Science 1839. Proceedings of 3rd International Conference on Intelligent Tutoring Systems ITS´2000* (pp. 534-543). Berlin Heidelberg: Springer Verlag.

Mislevy, R. J., & Almond, R. (1997). Graphical Models and Computerized Adaptive Testing. *Center of the Study of Evaluation (CSE)*.

Olea, J., & Ponsoda, V. (1996). Tests adaptativos informatizados. In J. Muñiz (Ed.) *Psicometría* (pp. 731-783). Madrid: Universitas.

Owen, R. J. (1975). A Bayesian Sequential Procedure for Quantal Response in the Context of Adaptive Mental Testing. *Journal of the American Statistical Association,* 70(350), 351-371.

QuestionMark. URL http://www.questionmark.com/us/home.htm [Accessed 2001, May 21].

Rasch, G. (1960). *Probabilistic Models for Some Intelligence and Attainment Test.* Copenhagen: Danish Institute for Educational Research.

Ríos, A., Conejo, R., Trella, M., Millán, E., & Pérez-de-la-Cruz, J. L. (1999). Aprendizaje automático de las curvas características de las preguntas en un sistema de generación automática de tests. In *Actas de la Conferencia Española para la Inteligencia ArtificialCAEPIA'99*.

Ríos, A., Millán, E., Trella, M., Pérez-de-la-Cruz, J. L., & Conejo, R. (1999). Internet Based Evaluation System. In *Open Learning Environments: New Computational Technologies to Support Learning, Exploration and Collaboration. Proceedings of the 9th World Conference of Artificial Intelligence and Education AIED'99* (pp. 387-394). Amsterdam: IOS Press.

Rudner, L. (1998). An On-line, Interactive, Computer Adaptive Testing Mini-Tutorial. http://ericae.net/scripts/cat/catdemo.

Thissen, D., & Mislevy, R. (1990). Testing Algorithms. In H. Wainer (Ed.) *Computerized Adaptive Testing: A Primer* (pp. 103-136). Hillsdale, NJ: Lawrence Erlbaum Associates.

TopClass. URL http://topclass.uncg.edu/ [Accessed 2001, May 21].

Van der Linden, W. J. (1998). Bayesian Item Selection Criteria for Adaptive Testing. *Psychometrika,* 63(2), 201-216.

Van der Linden, W., & Hambleton, R. (1997). *Handbook of Modern Item Response Theory*. New York: Springer Verlag.

VanLehn, K., Ohlsson, S., & Nason, R. (1995). Applications of Simulated Students: An Exploration. *Journal of Artificial Intelligence and Education*, 5(2), 135-175.

Wainer, H. (1990). *Computerized Adaptive Testing: a Primer*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wainer, H., & Mislevy, R. (1990). Item Response Theory, Item Calibration and Proficiency Estimation. In H. Wainer (Ed.) *Computerized Adaptive Testing: A Primer* (pp. 65-102). Hillsdale, NJ: Lawrence Erlbaum Associates Publishers.

Webassessor. URL http://www.webassessor.com/index.htm [Accessed 2001, May 21].

WebCT. URL http://www.webct.com [Accessed 2001, May 21].

Weber, G., & Specht, M. (1997). User Modeling and Adaptive Navigation Support in WWW-based Tutoring Systems. In *Proceedings of the 6th International Conference on User Modelling UM'97* Vienna, New York: Springer.

Weiss, D. J. (1982). Improving Measurement Quality and Efficiency with Adaptive Testing. *Applied Psychological Measurement,* 6, 473-492.

Welch, R., & Frick, T. W. (1993). Computerized Adaptive Testing in Instructional Settings. *Educational Technology Research and Development*, 41, 47-62.

Appendix A: Student questionnaire

1. What is your previous experience with this subject?
   a) Only lectures
   b) Lectures and some personal work
   c) A lot of personal work
2. How much time did you need to learn the functioning of the SIETTE system?
   a) More than ten minutes
   b) Between five and ten minutes
   c) Less than five minutes
3. Do you consider the system useful in the learning process? (select only one)

   Not at all Very
   a b c d e

   Please specify the reasons:
4. Will you keep on using SIETTE to study?
   a) Yes
   b) I do not know
   c) No

   Please specify the reasons:
5. Would you recommend SIETTE to other students?
   a) Yes
   b) I do not know
   c) No
6. Do you find the interface easy to use? (select only one)

   Not at all Very
   a b c d e

   Please specify the reasons
7. What do you like most about the SIETTE system?
8. What do you like least? What improvements could be made?
9. Have you detected any mistake?
   a)Yes
   b) No

   Please specify the mistake:
10. Do you consider that the qualification the system gave you was... (select only one)

    Totally unfair Totally fair
    a b c d e
11. Do you consider that the questions the system posed were... (select only one)

    Very difficult Very easy
    a b c d e

    Please provide the following information

Obtained qualification:
Expected qualification:
Number of questions:
Previous experience:

12. Would you like to be evaluated with a traditional paper and pencil test or by the SIETTE system?
    a) SIETTE
    b) Traditional paper and pencil test

Thank you very much for your cooperation. Additional comments are welcome.