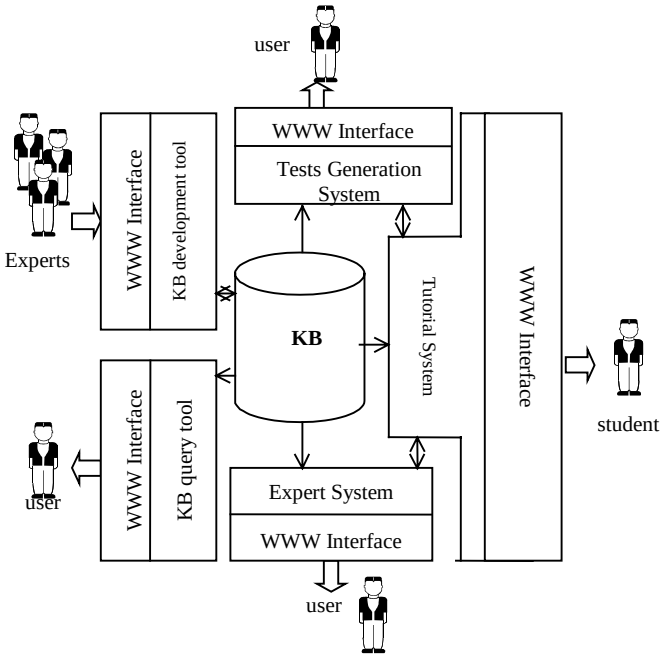
SIETTE: Intelligent Evaluation System using Tests for TeleEducation.

A. Rios J. L. Pérez de la Cruz R. Conejo
Languages an Computer Science Department
Málaga University
Tech Complex, Teatinos Campus 29071
Málaga, Spain
E-mail: {rios, cruz, conejo}@www.lcc.uma.es

ABSTRACT

This workshop is included in the TREE project, subsided by the LEONARDO program of the European Union, whose finality is the development of an Intelligent Tutoring System (ITS), accessible via World Wide Web, which will help in the classification and identification of different European vegetable species. An evaluation system using adaptive testing is under development to be used by the ITS diagnostic module.
The testing generation system has a complete set of tools that allows students self-assessment with accuracy based in his proficiency level and helps teachers in test development and design as well as to identify students with problems creating *learning profiles*.
This system is provided with a *temporary student model*. This model allows using the system as an independent tool over the WWW (SIETTE system), or as part of the diagnostic module in any ITS which want to give feedback to the students or estimate their initial proficiency level in any subject (TREE module). Our ITS uses a Knowledge Base structured by curriculums.
The goal is to join the dynamic nature of computer adaptive testing with a temporary student model and the advantages that offer the WWW as a learning tool and therefore as an evaluation method.

INTRODUCTION



The system described in this paper is part of the TREE project (Training of European environmental trainers and technicians in order to disseminate multinational skills between European countries). This project is included in the EU Leonardo da Vinci Program whose final aim is to develop an ITS which will help in the classification and identification of different European vegetable species.
The test generation system (SIETTE) is one of three tools that form part of this project as shown in the figure 0.

In this figure we can see that the main modules are: an ITS, an Expert System (ES) and an Test Generation System (TGS).
These tools use a Knowledge Base (KB) about the botanical domain. The ES, the ITS and the TGS use a specific KB and independent of the system KB. All these modules are accessible via WWW, including the KBs. Each one of them has a creation and update interface.
The TGS used in TREE is being developed and implemented as an independent and reusable system for the design and generation of adaptive test over the WWW. Also, the system can interact with any ITS that has a KB structured in curriculums and a student model defined as an

semantic network where each node is a curriculum's component with an knowledge level associated to it.
The advantages of using the WWW in the educational environment are: Multimedia content, hypertext capabilities and client/server architecture that allows distance learning.
The objective of the system described in this paper is to join the dynamic nature of computer adaptive tests and the advantages that offer the WWW as learning environment and therefore as an evaluation method.
Traditional test evaluation methods depend on the static process to store and analyze the data. On the other hand this process is dynamic in adaptive tests. So, an adaptive test is a test where the presentation of each item and the decision to finish the test are dynamically adopted based on the student proficiency. If we add a temporary student model, we get that the adaptive test will generate descriptive information about student learning mode and therefore it is possible to identify students with learning problems. Its elaboration is similar to the classical pen and paper tests with the difference that in adaptive test the questions posed to each student depends on his actual proficiency level. To achieve this goal it is needed to calibrate the possible test questions with parameters that allows to determinate the question that correspond to each student at any time.
One clear advantage of adaptive testing against traditional testing is that they can effectively limit the total testing time, because the next question posed to student depends on the previous answered questions. Another advantage is that computers are more exact and accurate calculating the student's score, and they can give a descriptive feedback in addition to a diagnostic based upon the student model.

SYSTEM ARCHITECTURE

The SIETTE system architecture embeds the main components of an adaptive test generation system in addition to a temporary student models, and group them in five modules: Questions bank, temporary student models, tests editor, adaptive tests generator and learning profiles generator. Graphical representation of this architecture so that the possible interactions with an ITS is shown in figure 1.

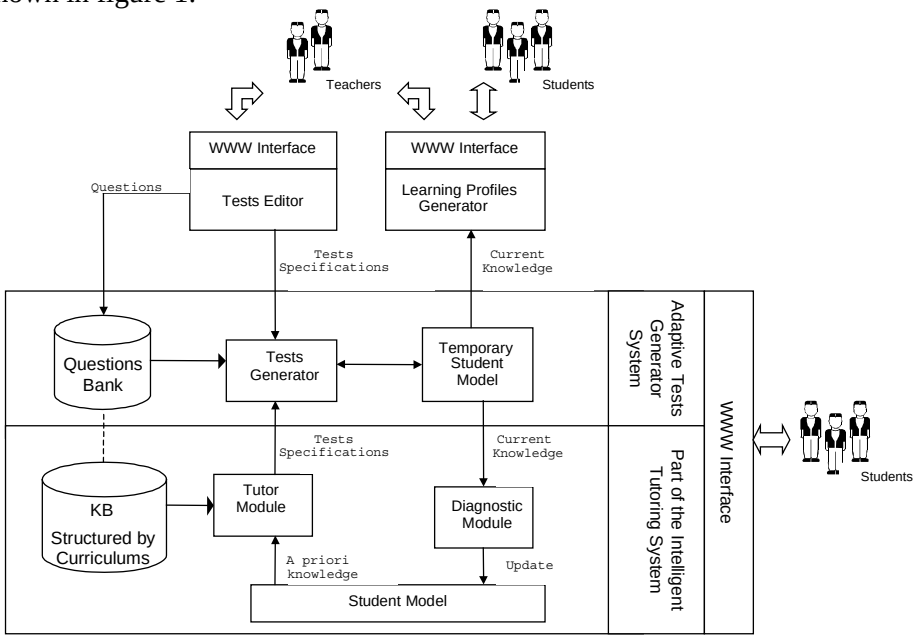


Fig 1. Interaction of the Adaptive Tests Generator System and the Intelligent Tutoring System

The *questions knowledge base* is a collection of possible questions to pose in a test. All these questions are calibrated with some parameters. It is built making use of the *test edition module*. This module allows the teachers to store questions and possible responses and to specify *the*

curriculum test. The *tests generator* is the main module of SIETTE. It's responsible of selecting the questions that are posed to each student, based on test specification and the *temporary model* of the student.
The temporary student model is a data collection required for SIETTE system to estimate with accuracy the real proficiency of a student on an specific subject of the whole curriculum. This model is used by the diagnostic module of the ITS in order to advance across the curriculum, to give advertisements, etc. The temporary student model is different from the student model of the ITS, which information is more complete. The temporary student model contains only information about the proficiency level of the student at a given moment of adaptive test.
To implement the questions bank and the temporaries students models we have used a relational database that can be accessed via WWW with scripts. The basic structure of a temporary student model can be:

Students						
StudentID	TestID	Date	Level of Proficiency	Lower Confidence Level	Upper Confidence Level	...

Knowledge's Distributions						
StudentID	TestID	Likelihood Level 0	Likelihood Level 1	Likelihood Level 9	...	Likelihood Level 10

Percents of Administrated Questions			
StudentID	TestID	TopicID	Percent

Posed Questions	
StudentID	QuestionID

Finally, the SIETTE system has a *learning profiles generator module* that gives information about each evaluated student. These profiles are available for the teacher and the student. Later in this paper we will see the information given by these learning profiles.

TEST EDITOR

Test designers can create the test specifications using the test edition module. Thus, the tests editor is a tool for extracting expert's knowledge using HTML forms. The supplied information is saved in a relational database.
In SIETTE, a test is organized in a structured manner in *subjects* and *questions*, related to each other by the existence of ownership relations defined with the editor. The relations that can exist between the subjects that form part of the course (aggregation and prerequisites relations) are defined by the ITS KB structured in curriculums. The existence of a curriculum of the course pretends that the questions selection algorithm use this structure and generate tests whose contents are balanced and follows the structure defined by the test specification [1][2][3]. Besides, the test designer is the person who calibrates the questions from the bank by means of parameters that will change dynamically as questions are posed to students [3]. This approach eliminates the need of the empirical previous study used by the IRT theory [4].
A possible course structure in the SIETTE system is shown in figure 2.

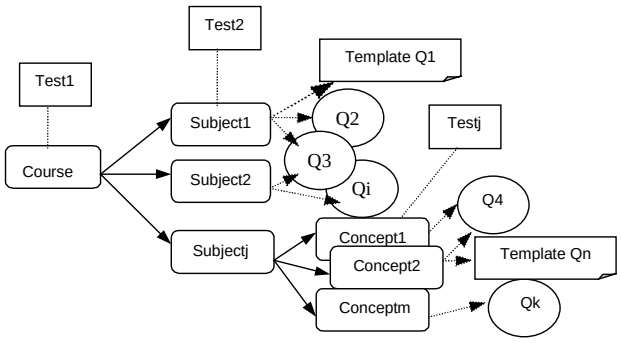


Fig 2. The system's views in different tests

The main advantage that offers this edition module is *components reusability*: the test from the same course can share subjects and these can share questions. Besides, these components can be written using HTML, with the flexibility that this language offers. Another advantage of the

tests curriculum is the capability of *multimedia content* in the questions and the possible responses. The limitations to the multimedia capabilities offered by the system are given by the HTML language limitations and actual browsers capabilities. With the inclusion of these contents, the number of subjects that can be evaluated with this system grows up against the classical current computer tests (static content). The mechanism for storing the multimedia content via WWW is based on RFC 1867 (Form-based file upload in HTML) [6].
The most important area to be developed in the SIETTE system is the possibility to define *questions and answers templates*, that is to extend the HTML language for editing tests instead of defining questions one by one. The user can define templates that will be dynamically instanced at run time if one of them is selected by the system. Template definition can give N possible questions (or answers) in the test. The system will randomly choose one of them. Therefore, template definition minimize the risk of repeating questions, students knowing the entire bank content and the teacher need of introducing a great number of similar questions in the bank.

TESTS GENERATOR AND EVALUATION ALGORITHM: QUESTIONS SELECTION, SCORING METHOD AND TERMINATION CRITERION

This module has been implemented using a CGI application implemented in C language[1].
The tests generator algorithm consists of three procedures: (1) Question selection based on the student model. (2) Proficiency estimation and update of the temporary student model. (3) Checking the termination criterion.
To select a question, we use Owen's Bayesian method. However, we postulated that student's knowledge is not infinite but it can be modelled as a finite set of discrete values, i.e. {0..10}. Thus, to calculate the level of student's knowledge, we only analyse, using bayesian method, part of the distribution that encloses that values [4]. Moreover, it facilitates the computation of the new knowledge's estimation and its confidence interval.
We improve that item selection procedure with some constrains. Constrains that have been added are:

1. Random item selection: One important characteristic in SIETTE is that the selected question among a set of candidate questions could be a template, so SIETTE has to do a new selection among a set of instances generated by this template. This selection is aleatory. (The same occurs with the possible answers).
2. Content Balancing: In SIETTE system, the desired content coverage of a test may be specified by the teacher, as the percentage of the test questions that should be chosen from each of the subgoal areas on the test. In this way, before selecting a question, we must select a subgoal, and for do it, the empirical percentages are compared to the prespecified desired percentages, and the subgoal with the largest discrepancy is selected.
3. Longitudinal testing. Item selection strategy in SIETTE, avoid administering the same items to a student who is tested more than once. Thus, it creates a record of the items administered in earlier testing, this record has an expiration date to indicate how long a student should go between presentations of the same item.

Like in the CBAT-2 algorithm [3], questions will change the difficulty level according to the number of times those questions are posed to theses students.
Once the better question have been chosen, the system poses that question to the student and waits for an answer. When the student responds the question, SIETTE system computes his/her new proficiency level and its confidence intervals.
With the new proficiency level, with its confidence intervals, and with the information about questions posed and the percent of passed subgoal, the system updates the temporary student model.

Then, the system checks the termination criterion that consists of three conditions: (1) The proficiency level has passed the confidence interval defined for the test designer. (2) Every

[1] An example of how a test is generated can be see in http://alcor.lcc.uma.es/siette

component has at least the minimum number of associated questions defined in the test specifications selected. (3) The system has posed the maximum number of questions defined for the test designer.
Once a test terminates, the temporary model of each student becomes its current knowledge. In this moment, students and teachers can query the learning profiles making use of the learning profiles generator tool, or in the ITS case, the diagnostic module updates the student model with this information.

SINGLE AND COLLECTIVE LEARNING PROFILES GENERATOR

This is the system tool that uses the temporary student model in conjunction with the testing strategies to facilitate educational decisions: shows a statistical paper that describes the changes in the temporary student model. Some techniques that we use come directly from ASRT [5].
In short, we can get a description about one test for each student and group of students: name and description of the test, number of questions posed, number of correct questions, subjects that have been passed, a comparison of each student with himself if he does the same test more times, the final knowledge's distribution, the final scoring (pass /no pass) with the level of proficiency and its confidence interval.
Both students and teachers can query these profiles. If the teacher wants to query the profiles, he can get the profile of one students or group of students that making the same test and in conclusion, to apply the most appropriate tutorial strategy.

CONCLUSIONS.

The system analyzed is an easy solution to the adaptive tests problems, at time that joins the dynamic nature of them with the advantages that the WWW offers as learning environment and therefore as an evaluation method. Using the WWW we can reduce the effort of evaluation of a big number of students. The evaluation is impartial and the results are more consistent and more accurate than with traditional paper-and-pencil tests. On the other hand, all tools of SIETTE can be accessed simultaneously, so a lot of people can do different things in base of their needs. SIETTE uses HTML-like language for editing questions, therefore the format and aspect of the questions are totally adaptable to teachers preferences. Generated tests can embedded multimedia objects (not only text), so teachers could compose better test interfaces and students may have a more attractive learning environment.
Finally, we would like to address that it's very important that the algorithms used to select the best question and to score the results should be efficient. Otherwise, the student might get bored and leave the test. We have to take into account that the time delay is due both to the internet transfer and to the algorithms efficiency. To improve the average performance it may be interesting to use the internet time delay to run the algorithms looking for the next question in the server side, while the student is still waiting or thinking the last question in the client side.

REFERENCES.

[1] Kingsbury and Zara (1989): Procedures for Selecting Items for Computerized Adaptive Tests. Applied Measurement in Education, 2(4), 359-375.
[2] Welch, R. and Frick, T. W. (1993): Computerized adaptive testing in instructional settings. ETR&D 41, 47-62.
[3] Huang, Sherman X. (1996): On Content-Balanced Adaptive Testing. Computer Aided Learning and Instruction in Science and Engineering, pp. 60-68.
[4] 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.
[5] Weiss D. J. and Kingsbury G (1979). An Adaptive Testing Strategy for Mastery Decision. Journal of Educational Measurement. Vol 21, Na 4, pp.361-375.
[6] Nebel, E. and Masinter, L (1995): RFC 1867 : Form based File Upload in HTML. http://sunsite.auc.dk/RFC/rfc/rfc1867.html