Abstract – The educational society faces the challenge of improving the quality of education, while increasing the percentage of students that pass a class, given limited teaching resources. In response to this demand, this work reports on the architecture and the implementation of a system for adaptive multi-user assessment of students. The software is a client/server platform developed in Java programming language, which is termed Platform for Adaptive and Reliable Evaluation of Students (PARES). Hypertext markup language is used for the development of pages including text and figures. PARES includes three modules, which can be remotely connected to the system Data Base on server through network. The Bayesian Decision model is applied to adaptive testing, instead of the traditional Item Response Theory. PARES can be used interactively by administrators, course instructors and students. When PARES is used for self-assessment, feedback is provided to the student’s answer. As a result, personalized learning assistance is provided to improve a student’s level of proficiency. Experimental results show that when students take self-assessment tests a week before scheduled formative tests, their scores are increased.

Index Terms – Computerized adaptive assessment, self-assessment

INTRODUCTION

In Computer Adaptive Testing (CAT) [1], tests are adjusted to the performance level of individual students. During testing, items are administered one at a time. The selection of each next item depends on the student’s response to the items previously administered. Adaptive assessment has additional benefits over Computer-Based Assessment [2], such as the accurate and reliable estimation of proficiency, the reduction of testing time, and the avoidance of very easy or difficult items that cause boredom or stress respectively.

Computerized Adaptive Tests are appropriate not only for formative or summative assessment, but also for self-assessment. It is recognized that self-assessment tests can be used for instructional purposes, when item correction is shown within some feedback [3]. Therefore, CATs can facilitate the integration of assessment with learning.

CAT is used as a diagnostic module to personalized e-learning environments, such as, KOD [4], INSPIRE [5], PEL-IRT [6], and TAPLI [7]. CAT is also used in Web-based tutoring systems, for instance, ELM-ART [8] and SEATS [9].

Traditionally, Item Response Theory (IRT) [10], a well-founded psychometric theory, has been extensively used as the underlying model in Computer Adaptive Testing. IRT has been used to determine the probability of a correct response, update student level or select the next item [3]-[9].

Adaptive mastery testing is used to classify students as masters or non-masters, based on their responses to adaptively selected test items. Various approaches on adaptive mastery testing have used Bayesian Decision Theory to select items and determine when to terminate testing. However, Decision Theory has been mostly used as a supplement to the IRT-based approach. Rudner [11] proposes simple Bayesian decision theory as an alternative underling model for adaptive mastery testing.

PARES – Platform for Adaptive and Reliable Evaluation of Students, is a software platform for automated student assessment, developed at the Department of Industrial Informatics of the Higher Technological Educational Institution of Kavala, Greece. The motivation under the development of PARES was the problem of the on-going, explosive expansion of higher education in Greece, given limited teaching resources. Under these circumstances, the challenge is to retain a certain quality of supplied education, while increasing ‘student throughput’, the latter is the percentage of students that pass a class.

Previous versions of PARES [12]-[13], enabled teachers to adapt tests manually. The current version incorporates MAAS, a Module for Adaptive Assessment of Students. The Bayesian decision theory is employed as an underling model for adaptive testing, in the way that Rudner [11] has proposed, with minor novelties. The adaptive modeling techniques implemented in the current version of PARES are briefly described in [14].

The emphasis of this paper is on the description of the architecture and functionality of PARES. The effectiveness of PARES is demonstrated comparatively in several classes. Applications of scale show that PARES can be useful for retaining a standard of education, while increasing students’ scores.

In the following section the theoretical basis of PARES is presented. Next, the system architecture is demonstrated and the details of how the application modules are used. Then, the experimental results are described. A discussion and the conclusions obtained from this work finalize the paper.

1 Catherine C. Marinagi, Department of Logistics, Technological Educational Institution of Halkida, GR-32200, Thiva, Greece kmarin@teikav.edu.gr
2 Vassilis G. Kaburlasos, Department of Industrial Informatics, Technological Educational Institution of Kavala, GR-65404, Kavala, Greece, vgkabs@teikav.edu.gr
3 Vassilis Th. Tsoukalas, Department of Industrial Informatics, Technological Educational Institution of Kavala, GR-65404, Kavala, Greece, vtsouk@teikav.edu.gr
THEORETICAL BASIS

Rudner [11] has used Decision Theory as a basis for adaptive testing within a latent class framework. The Bayes principle assumes that the prior probability of a student’s level of performance is available, before testing. Each time a single item is answered, a new student’s level estimate is calculated (a posteriori) using the item response and the prior probability values. Testing is continued as long as there is not enough information to make classification decision. After selecting an item that maximizes information, the estimated classification probabilities are updated in order to evaluate whether the test must be terminated or be continued.

The Module for Adaptive Assessment of Students (MAAS) [14] is incorporated in PARES to enable students to take adaptive tests. MAAS employs Bayesian decision-making with initial parameter values induced ‘on-line’ during self-assessment. The learning state of an individual student is represented by a vector of probabilities, the latter are used to determine the cost of selecting an item and then compute a customized ‘focus-of-attention’ vector of probabilities. The focus-of-attention vector, in turn, is used to draw stochastical questions from an Item Bank which (questions) are fed back to an individual student with the objective to improve a student’s learning state incrementally.

MAAS also combines adaptive item selection with random item selection, considering content-balancing percentages [15]. Random item selection requires items with the same level of difficulty, while adaptive item selection requires items which do differ in their level of difficulty.

THE ARCHITECTURE OF PARES SOFTWARE PLATFORM

PARES software tool is a device-independent, multi-user, menu-driven application, developed in Java programming language, which provides the advantage of easy installation in various environments. The architecture of PARES is a generic client-server or 2-tier architecture. The system DataBase is located on the DBMS Oracle DataBase Server 9i.

System application is placed on PC clients. Java Runtime environment (JRE), provided free from SUN, is appropriate for the installation of the application on PC clients. The application consists of three modules: the Administrator Module, the Instructor Module, and the Module for Adaptive Assessment of Students (MAAS). Each module can be used interactively by the respective type of user: administrator, course instructor, and student. PARES development in modules provides independence of the users’ work location and facilitates maintainability.

The system administrator installs the system DataBase on an Internet-accessed server. A username and password provide user’s authentication. Users can install modules on machines (PC clients), which may be remotely connected to the system DataBase on server through network, giving the IP address or the domain name of the server. A fast DSL connection is appropriate to ensure a satisfactory response to DataBase queries.

Figure 1 illustrates the proposed architecture of PARES software platform. Although just three PC clients are depicted, here, actually many machines can be connected simultaneously to the system DataBase. The remote DSL connections that are shown, can also be local connections of client machines located in the computer lab.

The DataBase contains:

- The Users Bank, which contains data concerning semesters, courses, instructors, and students.
- The Item Bank, which is organized hierarchically in a tree structure of chapters, units and items.
- Properties and rules for test composition given by the instructors.
- Students’ profiles and values for item parameters, which are dynamically re-estimated during adaptive testing.
- Tests dynamically created during assessment.
- Data kept during testing, such as students’ answers on tests, time until submitting a test, etc. These data are used for the re-estimation of parameters and statistical analysis.

PARES utility is maximized, since the Oracle DataBase Server and the client/server application do not depend on a particular operating system platform. Moreover, the environment where the PARES client/server application could be installed, does not depend on the environment where the Oracle DataBase Server could be installed.

SYSTEM COMPONENTS

PARES is a user-friendly tool, which consists of three modules, used by three different types of users.

An administrator uses the Administrator Module to give ‘access privileges’ to both instructors and students. A course instructor uses the Instructor Module to update the Item bank
and design tests. Students use the Module for Adaptive Assessment of Students (MAAS) to take either a test published by the course instructor, or a self-assessment test. The modules can be installed either on a client machine at the computer lab, or on any workstation that has network access to the server. We describe PARES modules and explain how they are utilized as follows:

I. Administrator Module

An administrator accesses Oracle DBMS with DBA privileges. He/she uses the Administrator Module in order to update records of semesters, courses, instructors and students, which are contained in Users Bank. Instructors are related with semesters and courses, and students are related to semesters, courses and tests. During a semester, students take a set of tests for each course. Their scores are recorded on the DataBase for statistical analysis. At the end of each semester, the final scores of each student for their courses are calculated and recorded on the DataBase.

II. Instructor Module

Firstly, course instructors use this module in order to update the Item Bank. The user-friendly authoring environment facilitates the usage of Instruction Module requiring minimal experience with PARES. In particular, the authoring environment contains:

- The Curriculum Editor, which is used for curriculum organization and item insertion.
- The Test Composer, which is used for test composition.
- The Test data Illustrator, which is used for the illustration of data gathered during testing.

Using the Curriculum Editor, an instructor organizes the curriculum of a course in a tree structure of chapters, units and items. Each unit contains items of the same content area and the same level of difficulty. Therefore, items of a particular unit are very similar. Each item includes a question, a set of possible answers and an associated text to each answer. The text provides feedback, i.e. a short documentation along with possible links to educational resources. The question type can be multiple-choice or multiple-response. Hypertext markup language (HTML) is used for the development of questions and answers. Therefore any object such as a graph, an icon or a mathematical type can be easily included.

All items can easily be imported from an external file or added manually, following the instructions of the wizard of Curriculum Editor. Although entering new items is straightforward, the instructors may consume time for the design of items, especially when specialized notation is used.

Figure 2 shows the window that enables the insertion of a question containing a scheme. Figure 3 shows the window that enables the insertion of an answer. The particular answer is the wrong one. The text containing documentation is associated with this answer. Links to URL addresses containing educational sources can also be included.

For each unit the course instructor assigns, by judgment, an initial level of difficulty of every item included, called Item Difficulty. The value of Item Difficulty will be re-estimated later, considering the responses of a group of students that take a test in order to initialize values of parameters.

Secondly, course instructors use the Test Composer of the Instructor Module to compose tests. Figure 4 depicts in Greek the Test Composer window for the course Software Engineering I. The window is divided in two parts. The left part depicts the tree structured Item Bank and the right part depicts the items (questions and answers) of the selected unit. Instructors choose from the Item Bank - clicking the corresponding icons - a particular course, the chapters and the units of each chapter that will be examined on a particular test. Then the instructor defines the test properties, which include the name of the test, a description, the range of scoring, an estimation of the time period needed for responding to the easiest and to the most difficult item, a choice of negative evaluation and a choice of adaptive or non-adaptive testing.

In order to produce content-balancing tests, the course instructor is enabled to predetermine the percentage of items that should be selected from each unit. At testing time, the items of a test are randomly selected from units of items.
considering these percentages. Content-balancing tests are easily produced in non-adaptive fixed-length testing, since the total number of test items is predetermined. In adaptive variable-length testing, content-balancing is ensured through a focus-of-attention vector.

For each subject, an instructor may store many composed tests, but authorize one of the test to be appeared on students’ screen before students sit for a test. At test time the instructor publishes the authorized tests on the students’ machines.

When all students have submitted their answers, the instructor automates scoring of answers. Data gathered during testing is recorded on the DataBase. The instructor uses the Test data Illustrator to have a graphical view of the data concerning students’ performance, or prints the evaluation report, which includes the list of students, who participated on the particular test and their corresponding scores.

III. Module for Adaptive Assessment of Students

The Module for Adaptive Assessment of Students (MAAS) is used by students to take scheduled tests or self-assessment tests. The module has three components:

- The Student Interface enables students to interact with the system.
- The Non-Adaptive Test Generator generates non-adaptive tests.
- The Adaptive Test Generator generates adaptive tests.

MAAS can be installed on a machine, which may be located in a laboratory of the department or may be remotely connected to the system DataBase on server. Students may enter laboratories in an asynchronous way, during the time that laboratories are reserved for tests. Students may participate in more than one test per time by entering the laboratory. They are seated in front of a computer screen and run MAAS, either when they need to participate in a scheduled test or when they need to take a self-assessment test to review their progress.

Pull-down menus enable students to connect to the server entering their username and password, choose the subject they want to be examined and choose between a scheduled test or a self-assessment test. When questions are appeared on screen (Figure 5), students give answers, submit tests, and then they may print preview, or print test report.

In case of a scheduled test, testing can be adaptive or non-adaptive. In non-adaptive tests, items are randomly chosen from the chapters or the units of the chapters that the course instructor has determined during test composition. A number of items are randomly selected from each unit, according to the predetermined content-balancing percentages of items per unit. If the Item Bank is large enough, the chances of each student receiving a different subset of items is potentially enhanced, thus attempting to cheat becomes pointless.

Each time a non-adaptive test is composed, the total test duration is dynamically calculated according to the time period needed to respond to each item. This time period corresponds to each Item Difficulty. When the test appears on screen, the timer on the screen starts to count down. In case that the time expires before submission, the answers are submitted automatically.

Self-assessment tests can be customized by course instructors for their own students by determining the chapters to be tested and assigning a content-balancing percentage of test items. An alternative is to allow students to choose the chapters they prefer from a self-assessment Item Bank.

During self-assessment, each student initially responds to a basic set of items and the student’s profile is estimated. Then, the test becomes adaptive. The next item that appears on the screen is adjusted to the student’s profile. When the student responds, the student’s profile is re-estimated and the
procedure is repeated until termination criteria are met. Students have access to any context and educational resources linked to an answer. Immediate feedback encourages better performance and fulfills the aim of learning. Figure 6, shows the appearance of a question that the student has already answered, the correct answer, the student’s answer and the comments on the particular wrong answer.

**FIGURE 6**
FEEDBACK GIVEN DURING SELF-ASSESSMENT

**EXPERIMENTAL RESULTS**

PARES has been successfully used for automated student assessment in the course *Software Engineering I*, taught at the Department of Industrial Informatics of the Higher Technological Educational Institution of Kavala, Greece. Table 1, summarizes the results of the real-world experiments. The first line reports the results of students’ performance using just paper-and-pencil tests during the academic year 2002-03. These results include only final scores and can be compared with the results of students’ performance using PARES in next three academic years (2003-2006).

Test parameters were kept similar throughout all experiments. The Item Bank included 180 items per course. Tests included 15 items on average, which were randomly selected from the Item Bank. The groups of students who participated in tests included about 60 per course on average.

During the academic year 2003-04, a non-adaptive fixed-length version of PARES was used. Teachers were enabled to adapt tests manually. Comparing with paper-and-pencil examinations of the previous semester, we conclude that PARES did not improve ‘student throughput’, which is the percentage of students passing a course. The reason behind this observation is that students were not familiar with PARES, they were not adequately prepared, and PARES was used only once in a semester. Students made complaints for the short test duration.

During the academic year 2004-05, a non-adaptive fixed-length version of PARES was used twice during a semester. However, this time a non-adaptive self-assessment test was administered a week before the scheduled formative assessment. Tests average scores and test range of scores were increased, compared to the previous year. No complaints were made by the students who participated. They enjoyed testing and PARES became a popular tool among their society.

During the academic year 2005-06, the last version of PARES, which incorporates MAAS for adaptive student assessment, was used. Formative assessment was non-adaptive, while self-assessment was adaptive. The experiment, which took place twice during a semester, included again self-assessment a week before formative assessment. The average test score and test range of scores were increased again. Students accepted PARES as an assessment tool and recognized that it helps learning in an enjoyable way.

These results show an increase of the ‘student throughput’, which is an important rate for both students and teachers.

<table>
<thead>
<tr>
<th>Academic Year</th>
<th>Version of PARES</th>
<th>Test score range</th>
<th>Average Test score</th>
<th>Test score Standard deviation</th>
<th>Final score range</th>
<th>Average Final score</th>
<th>Final score Standard deviation</th>
<th>Student Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-03</td>
<td>Paper-and-pencil tests</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>[0.7-9.0]</td>
<td>5.1</td>
<td>2.1</td>
<td>65%</td>
</tr>
<tr>
<td>2003-04</td>
<td>PARES tests : non-adaptive, fixed-length, randomly selected items</td>
<td>[0, 6.5]</td>
<td>2.4</td>
<td>1.57</td>
<td>[1.8-8.6]</td>
<td>5.4</td>
<td>1.4</td>
<td>67%</td>
</tr>
<tr>
<td>2004-05</td>
<td>PARES: same version as previous year. A non-adaptive self-assessment test is given a week before each scheduled test.</td>
<td>[0, 9.0]</td>
<td>3.85</td>
<td>2.2</td>
<td>[1.0-9.5]</td>
<td>5.2</td>
<td>2.5</td>
<td>73%</td>
</tr>
<tr>
<td>2005-06</td>
<td>PARES incorporates MAAS. An adaptive self-assessment test is given a week before each scheduled test.</td>
<td>[0, 10.0]</td>
<td>5.5</td>
<td>2.0</td>
<td>[2.5-10.0]</td>
<td>5.6</td>
<td>2.2</td>
<td>75%</td>
</tr>
</tbody>
</table>
This work describes an architecture for an adaptive assessment tool, termed PARES, which has been successfully used under real world conditions, at the Department of Industrial Informatics of the Higher Technological Educational Institution of Kavala, Greece.

Firstly, PARES can help instructors to diagnose students’ preparedness for a subject using either non-adaptive or adaptive testing. Secondly, PARES can be used as a self-assessment tool, to help students gradually improve their level of proficiency. The goal of self-management of learning can be met as long as students have the opportunity to access feedback associated with each item. Experimental results show that when students take self-assessment tests before scheduled tests, there is an increase in 'student throughput', the latter is defined here as the percentage of students that pass a class.

PARES consists of three application modules: the Administrator Module, the Instructor Module, and the Module for Adaptive Assessment of Students (MAAS). Three types of users: administrators, course instructors, and students can use modules interactively through a user-friendly interface. PARES development in modules provides independence of the users’ work location and facilitates maintainability. PARES is a client-server 2-tier application which works effectively for groups of up to one hundred students querying the DataBase simultaneously during an examination. The most implemented adaptive testing tools, either commercial [16] or academic [3]-[9], are Web-based. Then a question is raised: would it be better to modify PARES to a Web-based application implementing a 3-tier architecture?

A 3-tier architecture reportedly improves performance for groups of thousands of users and improves flexibility when compared to the 2-tier approach. A limitation with 3-tier architectures is that the development environment is more difficult to use and it is much more difficult to program and test software because more devices have to communicate to complete a user's transaction [17]. This is a problem when new methods are developed and implemented. As long as PARES is a pilot application, where new methods are tested and dynamic extensibility is required, a 2-tier architecture is preferred. In particular, this architecture facilitated the development, implementation and pilot application of MAAS, which adopts the Bayesian decision model to adaptive testing. A number of innovations on this model were successfully applied and tested thoroughly.

PARES can be easily modified to a web-based application. The three implemented application modules use hypertext markup language (HTML) pages. Both companies, Sun (for Java) and Oracle (for DBMS), provide application servers, as well as XML support for heterogeneous applications. In such a case, PARES can not only work as an independent assessment tool, but can also be integrated as a diagnostic module to personalized e-learning environments.

ACKNOWLEDGMENT

This work has been supported in part by the third European framework programme: Operational Programme in Education and Initial Vocational Training II.

REFERENCES