

Using Mobile and Web-Based Computerized Tests to Evaluate University Students

CRISTOBAL ROMERO,¹ SEBASTIAN VENTURA,¹ PAUL DE BRA²

¹*Department of Computer Science, Cordoba University, Campus de Rabanales, Edificio C2, Cordoba 14071, Spain*

²*Department of Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands*

Received 1 October 2007; accepted 15 January 2008

ABSTRACT: Mobile learning and testing is emerging as a potential educational environment. In this article we evaluate the use of mobile devices for testing as compared to web-based assessment systems. We also describe an authoring tool to develop adaptable and adaptive computerized tests that can be executed on such different platforms as personal computers, personal digital assistants and mobile phones. We have carried out an experiment with computer science university students to determine their satisfaction and to compare the results obtained when executing a test on personal computers versus mobile devices. The experiments have shown that students were highly motivated and enjoyed using mobile application for testing. Furthermore, there were not any significant differences in the results obtained with the different versions of the test. © 2009 Wiley Periodicals, Inc. *Comput Appl Eng Educ* 17: 435–447, 2009; Published online in Wiley InterScience (www.interscience.wiley.com); DOI 10.1002/cae.20242

Keywords: computerized test; assessment; adaptive test; mobile learning; web-based education

INTRODUCTION

Computer-assisted assessment or quizzes are among the most widely used and well-developed tools in education [1]. The main goal of testing is to measure the students' level of knowledge with respect to one or more concepts or subjects. Some examples include

state driving license exams, military training exams, job application exams in the private sector, entrance exams in post-secondary education, and certification exams conducted by and for professional groups. The use of assessment systems is also becoming more common in higher education institutions [2]. One of the main reasons for this success is the rapid advancement of Internet technology. Due to this, we can establish two main types of computer-assisted test systems.

Correspondence to: C. Romero (cromero@uco.es).
© 2009 Wiley Periodicals Inc.

- Computer-based testing (CBT) that simply uses personal computers (PCs) as the medium for students to take exams, for teachers to prepare tests, and for the transmission of traditional paper and pencil tests (PPTs) [3]. Compared to PPTs, CBTs offer logistical advantages, a unified platform for media, diverse item types, the tracing of test-taker actions, and immediate feedback.
- Web-based testing (WBT), which is thus named to distinguish it from the more conventional CBT which only works on a single PC or in a local area network. A WBT is a computer-based test delivered via the Internet and written in HTML language (HyperText Markup Language) and possibly enhanced by scripts.

However, a new type of Mobile test [4] has arrived with the proliferation of wireless handheld and mobile devices. m-Learning (mobile learning) and u-Learning (ubiquitous learning) have started to emerge as potential educational environments [5]. Mobile learning can be defined as e-Learning using handheld devices, such as personal digital assistant (PDA), mobile (cellular) phones, and smartphones (mobile phones with some extra functionality resulting from combining a PDA with a mobile phone). Mobile phones and PDAs have become very popular among students in our universities and they can complement e-Learning by creating an additional channel of assessment. New assessment systems have been developed or some existing ones have been adapted for use in mobile devices [6]. However, designing this type of applications that exploit new multi-platform technology is often a difficult problem [7]. Nonetheless, technological advances and standards can solve some of these problems and the highly personalized nature of digital mobile devices also provides an excellent platform for the development of learning systems [8]. Nowadays these have emerged as a potential platform for CBT due to such characteristics of the new generations of mobile devices as larger and higher resolution screens, more computing power, programmability in Java, or other languages, etc.

In this article we describe an authoring tool to develop web-based and mobile computerized tests. Our main objective is to develop a framework for authoring one single time and then delivery to different devices and platforms. Our second objective is to evaluate the students' satisfaction and results when they execute tests on PCs versus mobile devices. Our article has been organized as follows: first, we describe Test Editor, an authoring tool to author a text and then deliver it to many platforms, next, we describe the different test execution engines; then we

present the experiments that we have carried out; finally, we draw conclusions and propose future work.

BACKGROUND

Assessment has always been a very important step in the learning process. However, compared to web-based learning in general, web-based assessment is a relatively new development [9]. Its different forms are motivated by different purposes [10]: exams (teachers can see if students have reached an appropriate level of knowledge), self-assessment (students can check how much they are learning) and questions or activities (teachers can provide proper feedback while teaching). There are different types of items or questions [1], such as: yes/no questions, multiple choice questions (MCQ), multiple response questions, gap questions, matching-pairs or association questions, ordering questions, hot spot-single marker questions, pull down list questions, pointing-questions, graphics and painting questions, code-execution questions, calculated questions, open-ended answers, etc. MCQ is one of the most frequently used question types [2].

There are also two main types of computerized tests depending on control algorithms: regular or classic tests and adaptive tests [11]. A regular CBT (R-CBT) or conventional (classic) test is simply a computer-supplied version of a PPT, in which the assessment items (or questions) are the same for all students. Conversely, test items in adaptive CBT (A-CBT) or computerized adaptive tests (CATs) are such that each examinee typically receives different questions whose number can be smaller than the number of questions needed in a classic test to assess the student's knowledge with the same certainty [11]. The main difference between CBT and CAT is the possibility of adapting to each individual student [10]. However very little effort has thus far been directed towards providing adaptation features (either adaptivity or adaptability) for the learner [5]. Systems can be adaptable, allowing the users to change certain system parameters and adapt the system's behavior accordingly. Systems can also be adaptive, changing their behavior automatically based on the system's assumptions about the users' needs. User adaptability and adaptivity have gained popularity on the World Wide Web under the more general notion of personalization.

Computerized Test Systems

Nowadays, there are several well-known commercial as well as free tools for developing adaptive and classic computerized tests. MicroCAT [12] is the world's first computerized adaptive testing system for

PCs and FastTEST [13] is the first Windows item banking software that performs some of MicroCAT's functions, developed by the Assessment System Corporation. QuestionMark [14] is a well-known and complete commercial assessment management system. Hot Potatoes [15] is a commercial suite for creating interactive quizzes. Webassessor [16] is a commercial all-in-one platform to manage assessments. TRIADS [17] is a powerful set of authorware assessment routines developed by the University of Liverpool, Derby and the Open University. SIETTE [10] is a web-based system for adaptive test generation developed by Malaga University. Test++ [18] is an adaptive training system on the Internet developed by Salerno University. QuizPack/Quiz-Guide [19] is a system for authoring and delivers web-based quizzes for programming-related courses developed by Pittsburgh University.

Some of these systems use an arbitrary XML (eXtensible Markup Language) format to record the information about assessments and some of them use IMS Question and Test Interoperability (QTI). QTI is an international (and XML-based) standard for specifying and sharing tests and assessment data. But all the previously described systems are designed to be used only on a PC and not on mobile devices.

Mobile Test Systems

There are several commercial and free quiz systems specifically oriented for mobile users. Moodle for mobile tests [20] is a module to execute Moodle questionnaires using CHTML (compact HTML) compatible mobile phones with i-mode technology developed by the Sapporo Gakuin University. Interactive test is a module of an m-learning system [21] specifically developed for use in Java-enabled mobile phones and is developed by Malaga University. CAT-MD [22] is a computer adaptive test for mobile devices developed by Macedonia University. Daifukuchō [23] is an assessment tool developed by Mie University that uses specific mobile phones with a QR Codes scanner. Mobile EMT-B quiz [24] is a commercial testing software only oriented to handheld devices compatible with PalmPrinter software. Go Test Go's [25] is a commercial quiz system designed to be used in Java mobile phones which provides a great selection of quizzes about science, history, sports, movies. Mobile quiz [26] is a commercial turnkey mobile gaming template for a question/answer service that is based on a SIM browser or Java technology. The Classroom Performance System (CPS) is a commercial system [27] that uses specific mobile devices for testing (different

types of infrared response units or clickers with buttons and with or without a screen to obtain immediate feedback from every student).

Although some of the above systems use XML files to record test information, all are designed to be used only on mobile devices.

Computerized and Mobile Test Systems

There are also some quiz systems that can be used both on PCs and mobile devices. Quizzler [28] is a commercial product designed for PC, Palm and PocketPC platforms and provides a collection of tools (quizzler reader, quizzler software maker and quiz library). M-Quiz [29] is an adaptive quiz system for a PC, PDA (iPaq), and mobile phone that uses Active Server Page (ASP) and is developed by the Victoria University of Wellington. CosyQTI [30] is a tool developed by the University of Piraeus for authoring adaptive assessments that can be accessed by desktop and handheld devices with WAP (Wireless Application Protocol) technology. C-POLMILE [31] is an intelligent mobile tutoring system (ITS) developed by the University of Birmingham that lets one execute MCQ both on desktop PCs and handheld computers.

However, none of these assessment systems can be used to develop and execute adaptive and adaptable tests in both web-based and mobile devices. In this article, we describe a complete test system that has all these characteristics and that has been integrated into the AHA! (Adaptive Hypermedia for All) system [32]. AHA! is an Open Source Web server extension to add adaptation to applications such as on-line courses. Users request pages by clicking on links in a browser, and AHA! delivers the pages that correspond to these links depending on three different types of information (domain model, user model, and adaptation model). AHA! was originally developed to support on-line courses with some user guidance through conditional (extra) explanations and conditional link hiding. But now AHA! has many extensions and tools that have turned it into an adaptive and versatile hypermedia platform and it has a complete set of authoring tools to allow authors to easily create or change applications or courses, concepts, concept relationships, computerized tests, etc. In short, we have used the AHA! system mainly because it is a well-known open source architecture used to build web-based courses, and because it uses Java and XML languages.

Table 1 compares the main characteristics of all the previously described quiz systems and of our system. As can be seen, there are no other quiz systems that offer all the features of our system.

Table 1 Main Characteristics of Some Assessment Systems

	Public tool	Author and report tool	Import tests	N. of type of items	Web-based execution	Adaptive test	Mobile execution	Adaptable interface
AHA!Test	Yes	Yes	Yes	3	Yes	Yes	Yes	Yes
CAT-MD	Yes	No	No	1	Yes	Yes	Yes	No
CosyQTI	Yes	Yes	No	4	Yes	Yes	Yes	No
C-POLMILE	Yes	No	No	1	Yes	No	Yes	No
CPS	No	Yes	Yes	2	No	No	Yes	No
Daifuku-cho	Yes	Yes	No	1	Yes	No	Yes	No
EMT-B quiz	No	No	No	5	No	No	Yes	No
FastTEST	No	Yes	Yes	6	No	Yes	No	No
Got Test Go	No	Yes	Yes	1	No	No	Yes	Yes
Hot Potatoes	No	Yes	Yes	8	Yes	No	No	No
Interactive test	Yes	No	No	2	No	No	Yes	No
MicroCAT	No	Yes	Yes	6	No	Yes	No	No
Mobile Quiz	No	Yes	No	1	No	No	Yes	No
Moodle Mobile	Yes	Yes	Yes	5	Yes	No	Yes	No
M-Quiz	Yes	No	No	1	Yes	Yes	Yes	No
QuestionMark	No	Yes	Yes	20	Yes	No	No	Yes
QuizPack	Yes	Yes	No	1	Yes	Yes	No	No
Quizzler	No	Yes	Yes	3	Yes	No	Yes	Yes
SIETTE	Yes	Yes	No	2	Yes	Yes	No	No
Test++	Yes	Yes	No	5	Yes	Yes	No	No
TRIADS	Yes	Yes	No	39	Yes	Yes	No	No
WebAssesor	No	Yes	Yes	5	Yes	No	No	No

ARCHITECTURE FOR DEVELOPING MULTI-DEVICE COMPUTERIZED TESTS

An assessment system should apply the Triple-A Model [33] as the baseline qualification in order to provide the most comprehensive form of test and to be more suitable for teacher education. In Figure 1, we can see the Triple-A model architecture that we have used in our system and that consists of three main functions:

- *Assembling*: to construct item pools and tests. Firstly, teachers have to create/select what course concepts will be evaluated by the test. Then, they have to create items (i.e. questions and their answers) and finally, group items into tests.
- *Administering*: to show the test to the students so they can execute it; to collect and record the score data. Students have to use a web browser to execute the test on a PC or they can download the test (.jar file) onto a mobile device and take the test while on the move.
- *Appraising*: to analyze the collected score data of a test and to generate a report with statistics. Teachers can select a specific test or item in order to generate a report. Using this information they can do maintenance tasks (for instance decide on

questions that are too difficult or answers that appear to be misleading).

As we can see in Figure 1, assembling and appraising are done in off-line mode while administering is in on-line mode. We have developed a complete framework consisting of a Test Editor (authoring tool) and several test execution engines. The proposed architecture enables tutors to author one single time and then deliver on both mobile and web-based platforms. We have used XML to store all the information (items, test files and also .aha course files are XML files) and Java language to execute tests on different devices. In short, we have developed two test engines to execute the same XML tests on desktop/PCs as on mobile phones and PDAs.

TEST EDITOR AUTHOR TOOL

Test Editor is an authoring tool for building computerized tests. We have created it in order to aid the teacher in the creation and maintenance of computer-based tests. Test Editor is integrated into the AHA! system, so it has a similar interface to other AHA! authoring tools such as Form Editor, Concept Editor, and Graph Author. The steps for developing

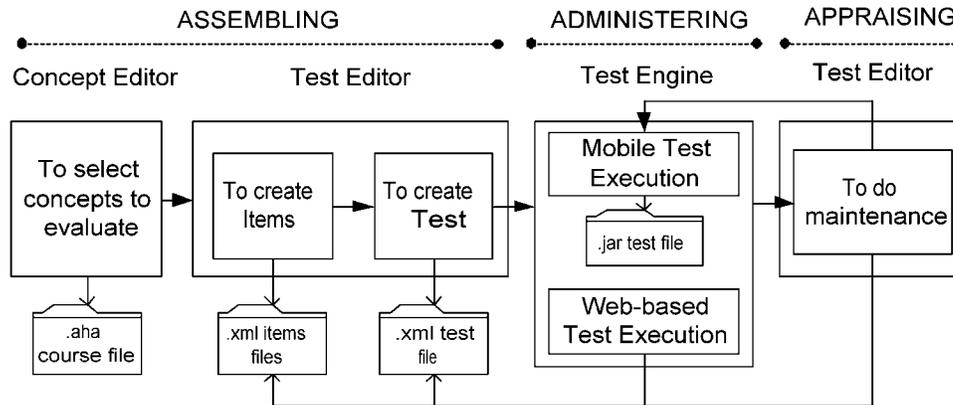


Figure 1 Main characteristics of some assessment systems.

and maintaining a test using Test Editor are found in continuation.

Creating an Item File

First, the examiners have to create one or more item files. Each item consists of a single question about a single concept (of an AHA! course), which they want to use to evaluate the students' knowledge. These concepts are created using AHA!'s Concept Editor or Graph Author. Several items about the same concept can be grouped together into a single item file. Test Editor provides three types of items:

- *True/false items*: where students must select just one answer as correct from two options.
- *Multiple-choice items*: where students must select one answer as correct from several options.
- *Multiple-response items*: where students must select one or more answers as correct from several options.

These types of items can be combined later in the same test. In order to build each item, the examiners have to specify some required parameters such as: the question (called "enunciate" in the item file), the answers and for each answer a flag to indicate whether the answer is correct or not. They can also specify some other optional parameters such as: an illustrative image (to upload from their local hard disk), explanations of wrong answers, and Adaptive or Item Response Theory [11] (IRT) parameters (item difficulty, discrimination and guessing). Then, items are stored in an XML file.

Examiners can use Test Editor to add, modify, delete, or create new items. They can also import and export items from/to other well-known computerized test formats and standards such as IMS QTI [34],

QuestionMark [14], Gift [35], XML-Moodle [35], Web-CT [36], Hot potatoes [15] and SIETTE [10]. In this way, teachers can re-use questions from other test environments without needing to enter them again.

Creating a Test File

The second step is to build complete tests out of items. The examiners decide on the test type they want (classic test or adaptive test) and whether to use just one or several item files. If the test evaluates only one concept, then it is considered to be an activity. If the test evaluates several concepts, then it is considered an exam about a chapter or perhaps a whole course. Next, the examiners can use different methods to select which specific items from the item files will be used in the test. This selection can be done manually (items are selected one by one), randomly (a number of items are automatically selected), or randomly with some restrictions (a number of items that fulfill some conditions). Then, examiners have to set the presentation parameters (see Fig. 2) about how questions are shown to examinees. For example, the order in which questions and answers are shown, whether to show or hide explanations about the answers, the maximum time to respond in and whether to show the correct answer for each question, and the final mark.

In addition to these parameters, there are some others concerning test evaluation and the web page. The examiners can determine how the test will be evaluated, for example, to penalize incorrect answers and to penalize unanswered questions, and what percentage of knowledge the final mark represents in associated concept/concepts. In addition, if the test is adaptive, the examiners have to set such adaptive algorithm parameters as question selection procedure and termination criterion. The examiners can also personalize the test interface, that is, the HTML page

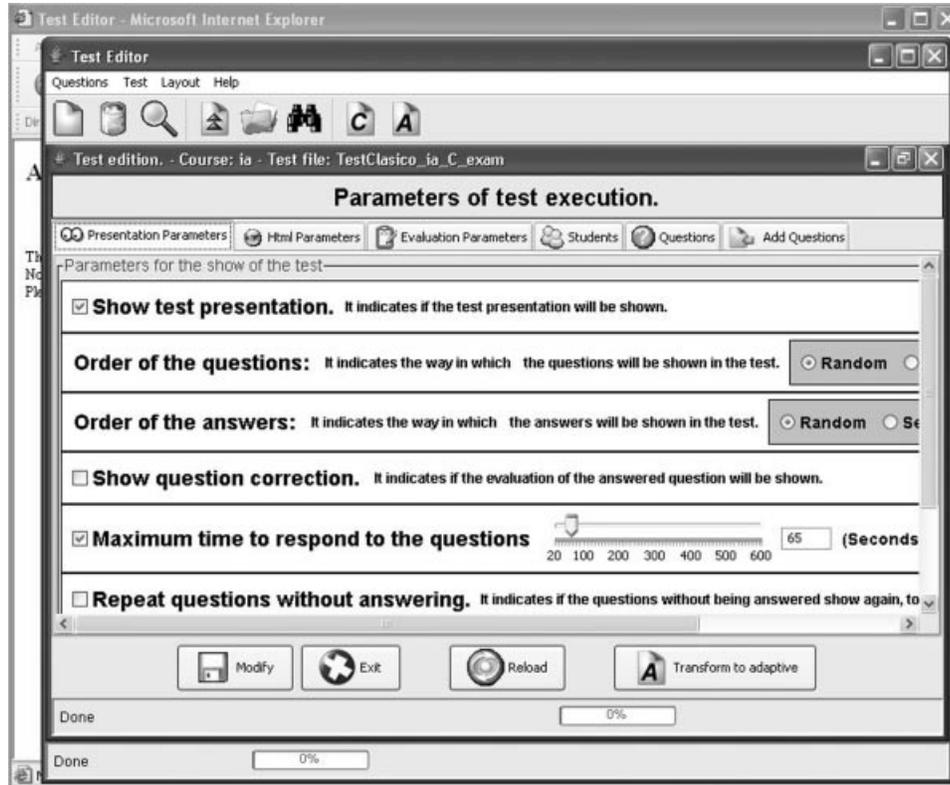


Figure 2 Test Editor window for selecting the presentation parameters of the test.

in which the test will be shown. They can select the background color, font type, and color of the text, and title of the page.

Finally, each test is stored in an XML file that can be read directly by our web-based test engines. We have also integrated this XML test file into a Java Archive (.JAR) file that includes both the mobile test execution engine as well as the test.

Executing and Maintaining a Test File

The third step is to have students take the test, using one of the two different test execution engines. If students want to use the web-based engine, they only have to open a web-browser, connect to the appropriate AHA! course and follow a link to start the test. On the other hand, if students want to use the mobile engine, they have to first download the .jar file onto the mobile device, then install it and finally execute it.

During test execution, our system stored a lot of usage information about the students and test. So, after a large number of examinees performed the test, examiners could obtain detailed (individual) and statistical information. Test Editor shows information about each student and about each item, for example, the success rate per question, mean time to answer the

questions and question usage percentage. Examiners can use this information for maintenance and improvements to the tests. They may decide to modify or delete bad items (too difficult, unclear, etc.), to add new items, or to modify the test configuration. Test Editor also can calibrate items [11], in order to directly transform a classic test into an adaptive one, or to optimize the IRT parameter of an adaptive test. Item calibration is to estimate test parameters. Traditionally, these parameters are estimated by teachers, but they can also be obtained from previous studies where many students are involved.

WEB-BASED TEST ENGINE

Our web-based test engine is a Java Applet. An Applet is an application that is embedded in a web page. In order to add a test to an AHA! course, authors only have to add a specific hyperlink to the XML test file on the corresponding course web page. Then, students using a browser have to be logged on to the appropriate AHA! course to get access to the test web page.

When a student starts a test (by clicking on the test link), the engine connects to the AHA! server in

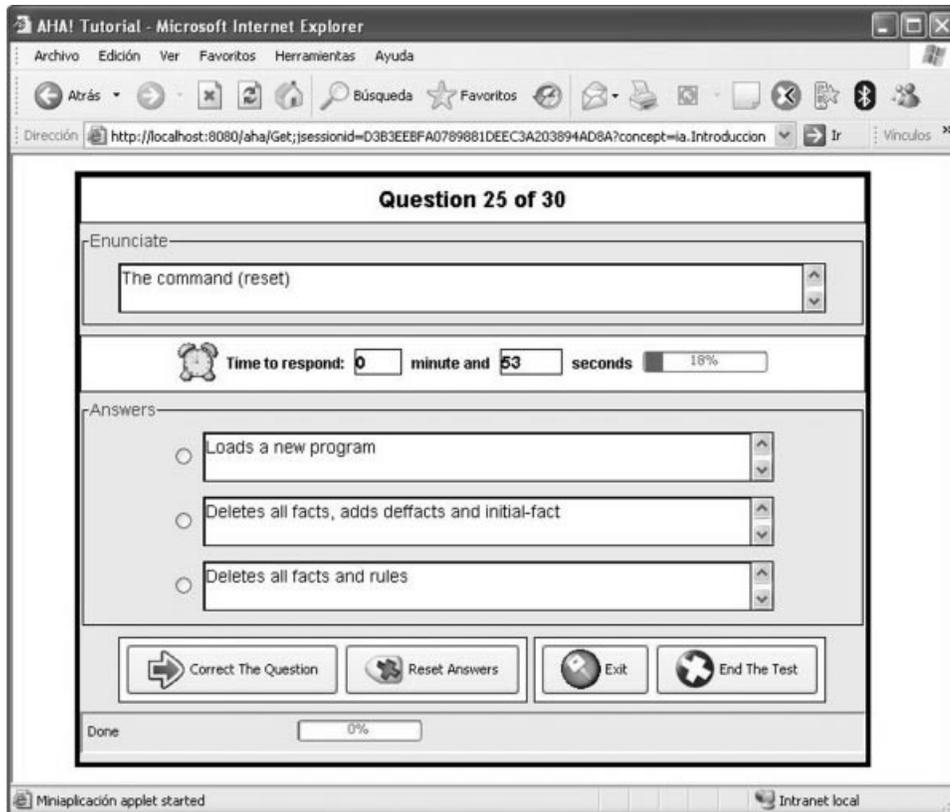


Figure 3 Interface of a question in a PC web-browser.

order to obtain all the test information. It also checks if the student is allowed to take the test or repeat it. Then, it starts to show questions one by one and the student only has to select what the hopefully correct answers are (see Fig. 3). This has to happen before the maximum response time has elapsed, if it is established. Then, students will see if the answer submitted was correct or incorrect, but only if the author has enabled it. After the students reply to the last question on the test, they will see the mark obtained and the total time elapsed, if the author has enabled this. Finally, the marks obtained will be automatically sent to the AHA! server where they are stored (centrally, for all users, in order to facilitate test statistics) and also to update the AHA! student's user model.

MOBILE TEST ENGINE

Our adaptable mobile test engine is a Java Midlet. Midlets are small applications designed to run on wireless Java enabled devices. Midlets can be used offline without connection costs and they provide interactive interfaces. Students have to install the test Midlet engine onto their mobile devices before they

can execute it. There are two different ways to do this: in remote mode, by downloading the .jad (Java Application Descriptor) file from Internet and installing the .jar file directly; or in local mode, by downloading the .jar file to a local PC first, and then sending it to the mobile device using Bluetooth, Infrared, serial bus, or other communication technology.

After installing the test Midlet application, students can execute it. The questions are shown on the device screen (see Fig. 4) like in the Applet version. They can be presented in linear or random order, or adaptive (depending on the test parameters). The students have to select the answers for each question using the phone keys and joystick (four direction cursor). Finally, when the test ends, the marks obtained and the time spent are shown and sent to the AHA! server.

Personalization and Adaptable Features

Our mobile test engine has some personalization features to allow individualized execution:

- When students start the application they have to identify themselves by introducing their



Figure 4 Interface of a question and final score in a mobile phone.

personal login and password (the same used in AHA!).

- Each time a student replies to a question, the answer is physically stored on the mobile memory card by using RMS (Record Management System).
- If a student executes an exam, then the elapsed time for each question is shown on the screen.
- Activities can be repeated several times by the same student, but exams cannot (students only can see the obtained scores).

Our mobile test engine also has some adaptable features in its interface. Adaptable systems are customized by the users themselves, allowing them

to adjust the interface or content provided by the system to accommodate their preferences [37]. In our Mobile test application, students can select the following preferences from the main menu (see Fig. 5 at the left):

- Students can select to what specific AHA! server the application will connect in order to send the marks obtained and to check if a particular student can execute a particular exam test.
- Students can select the language used in all texts of the interface. Currently, they can choose between Spanish and English.
- Students can choose between different font types (see Fig. 5 in the middle) and sizes, in order to improve the readability of the text in

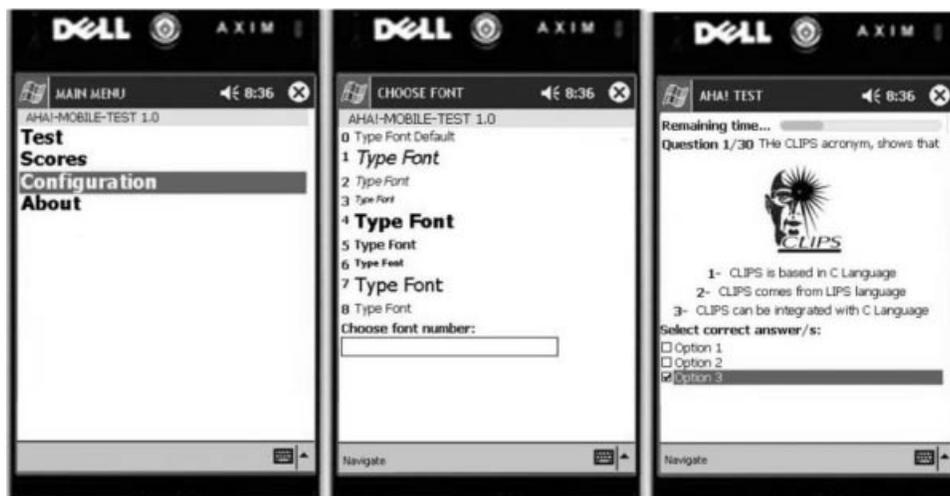


Figure 5 Interface of the main menu, font selection, and a question in a PDA.

the questions on a particular mobile phone or PDA.

- Students can choose to show questions and answers together on the same screen, for example if they have a PDA with a sufficiently large screen (see Fig. 5 in the right) or a mobile phone that allows them to scroll (vertically). Or they can choose to show questions and answers on two different “pages” or “screens” (if they prefer to see the question on one screen and the answers on another).

All these preferences are stored on the memory card of the mobile phone or PDA, so that if the user exits and comes back later to the test, all this information is remembered by the application. Most of these parameters have specific default values (default AHA! server, Spanish language and Arial font type). However, the font size and the number of screens in questions are adapted automatically by default depending on the size of the device screen in the following way: if the screen width is less than 128 pixels, then the font size is set to small and the number of screens is set to two. Or if the screen width is greater than 128, then the font size is set to medium and the number of screen is set to one. The size of the images (that are associated with some questions) is also adapted automatically to the screen size of the mobile device. This is very useful if the device does not have a screen large enough to show the original resolution of the images.

Security Issues

There are some security issues that we have taken into account to be able to use mobile devices to evaluate students. Our objective is to prevent students who wish to cheat from easily hacking and downloading a .jar exam.

- Students can try to cheat by uninstalling and installing the application in order to take an exam again. But our mobile application connects to the AHA! server every time that an exam starts in order to check that the user has started or terminated that exam before.
- Students can also try to turn off the mobile device in order to start the test again (or the mobile may turn itself off when the battery is depleted). In this case, our mobile application continues for the remainder of the test (the point where the test was interrupted is stored on the memory card).
- Another special case is if the students end the test but they have not connected to the AHA! server. Then the final mark and time taken are also stored on the memory card and when the device finally connects it will automatically send the results to AHA!
- Students can also receive a phone call or can try to switch between the test and other programs embedded in the mobile device. In this case, our mobile application is not paused and so, the testing time is still running.

But as in traditional paper-based or computer-based test exams, it is impossible for the teacher to guarantee that students do not receive a phone call, switch between applications or use paper notes during the test if the test is executed with students on the move rather than in a controlled laboratory setting.

J2ME and General Limitations

In order for a mobile device to be able to execute a Midlet, it needs to support the Java 2 Micro Edition (J2ME) virtual machine, and the application (.jar file) should be installed before running. But today almost all modern mobile phones and some PDAs are shipped with J2ME and it is very popular thanks to Java-based games [21]. Additionally, we have also found some general limitations for mobile devices that we have had to resolve:

- *Small screen size:* most of the mobile phones have a small screen, not always large enough to display a question and the possible answers at once. Students have to scroll in order to read the question, answers, and the remaining time. In order to resolve this, we have developed an adaptable configuration (which we have described previously) for adapting the test interface to different screen sizes.
- *Limited application size:* some mobile phones have a maximum Java .jar file size that varies from one model to another. In order to resolve this, we have tried to reduce the number of lines in the source code of the mobile test engine. In addition, we recommend that if the test has images, .jpg files with low resolution and high compression should be used.

EXPERIMENTAL RESULTS

We have carried out experiments with university students. Our objective is to evaluate students' results

Table 2 Comparing PC, PDA, and Mobile Phone Environments

	PC	PDA	Mobile phone
Input methods	Full keyboards and mouse	Touch screen stylus/pen	20 Keys and joystick
Output screen	Extra large (15" and 17") monitor	Medium (3" and 3.5") display	Ultra-small (1.5" and 1.7") and small (2" and 2.2") display

and opinion/satisfaction when they take tests on PCs versus mobile devices (mobile phone and PDA). We want to see whether there are significant differences in the results obtained and the opinions of students about to execute a test on three platforms. In order to evaluate the performance of the student, we compare the time taken to complete the test, the number of correct items, number of incorrect items, and the number of unanswered items. And in order to know what their satisfaction and opinions are, we have carried out a survey with specific questions and also personal comments.

Description of the Experiments

We have used three different platforms: PC (university computers), mobile phones (students own cell phones), and a PDA (which we have provided). PCs had Windows XP, Internet explorer, and the Mozilla browser with Sun's Java Plug-in 1.5. Different mobile phones were used: Nokia phone models (6xxx, 7xxx, and Nxx) with Symbian series 60 and with Sun's Java virtual machine micro edition. There were different PDAs: PDA Pocket PC models (Dell Axim, HP iPaq, Acer Cxxx, and Asus Pxx) with Windows Mobile and IBM J9 or Esmertec Java virtual machine micro edition. The main differences between the three devices are the inputs (input methods used) and output (the size of the screen) as we can see in Table 2.

We selected 30 computer science engineering students at Cordoba University, all about the same age (about 20 years old), with similar experience (second course of computer science technical engineering), using a Java enabled mobile phone and possessing basic knowledge about using a PDA. In this way, all of them were familiar with the use of the three devices (PC, PDA, and mobile phone).

The three tests consisted of 30 different items about the CLIPS (C Language Integrated Production System) language. One single teacher created the 90 items, each with 3 possible answers of which, 1 was correct. All the items had a similar level of difficulty and the number of words in each question was between 20 and 100. Finally, the exam was configured to show the questions in random order and the students had 1 min to respond to each question.

Performance of the Students

The three tests were executed sequentially (with an interval of 30 min) by the 30 students in the same classroom (at the university computer laboratory) in June 2006. Firstly, they execute the PC test, then the PDA test, and finally the mobile phone test. Before starting the PDA and the mobile phone test, the students themselves uploaded the exam to their mobile device from the teacher's PC. The examiner gave clear instructions to them for uploading the .jar file using Bluetooth and they did not have any difficulty.

Table 3 shows the mean value and the confidence interval (95%) of the time taken (in seconds) to complete the test, the number of correct items, number of incorrect items, and the number of unanswered items. In general, there are not many differences between the results obtained on the three platforms (see Table 3). A statistical analysis has also been done in order to reveal whether the three platforms do actually differ. We have applied an ANOVA for multiple comparisons testing: Tukey's HSD (Honestly Significant Difference) and REGWQ (Ryan-Einot-Gabriel-Welsch) tests show that only the time factor between the PC and Mobile test differed significantly. We can see in Table 3 that the

Table 3 Students Tests Execution Results: PC Versus Mobile and PDA Test

	Time taken	Correct items	Incorrect items	Without answer items
PC Test	764.4 ± 56.4	19.8 ± 1.1	6.2 ± 1.2	4.1 ± 1.2
Mobile Test	769.1 ± 53.2	18.9 ± 1.2	6.8 ± 1.7	4.9 ± 1.8
PDA Test	774.2 ± 66.3	19.3 ± 1.4	6.5 ± 1.3	4.4 ± 1.0

execution of the PC test is a little quicker than that of the Mobile test, and the PDA test is slower than the Mobile test. It is interesting to see that students browse fast (similar to PC browsing) through mobile interfaces (especially with mobile phones). But it is not very surprising if we think that some of the most popular software applications used by students are mobile applications such as Java games and SMS (Short Message Service) applications. Finally, we can also see in Table 3 that the final marks were very similar in the three versions with only small differences (PC test scores are the best, closely followed by PDA and Mobile test ones). It is logical that the scores obtained for the three platforms were similar because all the students had similar characteristics and knowledge.

Opinion of the Students

We have also carried out a survey among all the students in order to determine their satisfaction and opinions regarding the three versions of the test. The questionnaire was designed to gather information on students' attitudes towards aspects of the mobile learning they experienced as well as towards some more specific aspects of testing. The questionnaire had four questions (How much do you prefer it? How useful is it? How much do you like the user interface? and How much do you like the data entry method?) We use a five-point Likert-type scale [38] for evaluating students' ratings on given statements. Students had to answer with a number that ranged between 1 (a little) and 5 (a lot) for each question.

In Table 4 we show the mean value and the confidence interval (95%) of the ratings on preference, usefulness, the rate of acceptance of the user's interface, and the data entry method. Tukey's HSD and REGWQ tests showed that only in the last two questions (about user interface and input data method) were there significant differences between the three platforms, and not in the first two questions (about preference and usefulness). We believe that this can be due to the fact that the first two questions are about more subjective and abstract concepts and are therefore more difficult to evaluate. In general, the PC test

receives the most favorable evaluation in almost all the questions (see Table 4). Students prefer to take an exam on a PC, and they consider that the PC has a better user interface and input data method. This can be because our computer science students are still a little more familiar or comfortable using PC interfaces and their data entry methods for this type of applications (assessments and grading). However, it is very interesting to see that students consider that PDA and mobile phone tests can be more useful than the PC test. Specifically, students like PDAs more than mobile phones because they have a better user interface (a bigger screen to see the full question with answers) and input data method (touch screen with a stylus pen).

Students' Personal Comments

Finally, students could also write some comments together with the questionnaire in a personal comments area where they indicate the main advantages and benefits of PDAs and mobile phones:

- *Poor PC availability*: students think that mobile devices can be very useful for testing when the number of students in a classroom is larger than the number of available PCs. If some students use their personal mobile phone or PDA, then it will not be necessary to split the students into several groups for assessment activities.
- *Practice in spare time*: students think that mobile devices would allow them to execute test activities during their spare time, for instance while traveling or waiting for public transport, etc. In this way, they can self-assess and better prepare for an exam when they cannot use a PC.

And students think that the main weaknesses of mobile phones and PDAs are:

- *Very expensive*: almost all the students think that PDAs and Java mobile phones are very expensive at the moment. They must get cheaper in order for most of the students to be able to afford them. Once affordable, mobile tests and

Table 4 Student's Opinion Questionnaire About PC, Mobile, and PDA Test

	Prefer	Useful	User interface	Input data method
PC test	3.94 ± 0.23	3.84 ± 0.40	4.21 ± 0.23	4.38 ± 0.25
Mobile test	3.69 ± 0.31	3.89 ± 0.29	3.79 ± 0.29	3.98 ± 0.36
PDA test	3.76 ± 0.27	3.94 ± 0.30	4.05 ± 0.25	4.16 ± 0.29

other m-learning tools will become really useful and usable in the classroom and in real life.

- *Tiny buttons on some mobile phones*: some students with big fingers had problems pressing the button desired. They would like mobile phones with larger buttons or other alternative input methods to become available.
- *Power and screen saver systems*: most of the PDAs and mobile phones attenuate or turn off the light of the screen if you do not do anything for a (variable) time. Some students became confused by the screen blackout while they were reading the text of a question or the answers.

CONCLUSIONS

In this article we have shown an authoring tool that resolves the problem of authoring assessments only one single time for delivery on very different platforms (desktop/laptop PCs, PDAs, and mobile phones). In order to do so, we have used XML for storing test information and Java for developing the system. We have also created two test engines that can execute not only classic tests but also adaptive tests in different multi-devices. The main advantages of our test system are: it is modular (concepts, items and tests are clearly separated), it is easy to use (it has a user friendly graphical interface); it facilitates test maintenance (it has statistical information and item calibration based on examinees' usage information), it uses adaptive tests (to adapt the content of the test to each student) and it has adaptable and personalization characteristics (to adapt the interface of the test to user preferences and mobile restrictions).

After the experimentation, the first impression was that there were not too many differences between the results obtained when executing the test on the three platforms. The second impression was that students liked the experience of using a mobile application to evaluate their knowledge in a specific area. They were generally highly motivated to use mobile technologies for testing and found it useful, especially for self-assessment. So, we have shown how PDAs and mobile phone can be used as a supplement in e-Learning for evaluating students by means of Multiple Choice tests, in which the students only have to select an answer.

In the future we want to do more experiments with other types of students with different backgrounds (such as history, medicine, chemistry or physics students). Our evaluation results may have biases caused by the examinees. In our experiment, all the examinees are computer science students and we

can assume that they are intimately familiar with computer systems and applications. So, we do not know what results and opinions we would obtain using students from other different backgrounds. We are also interested in studying the opinions of the teachers. We want to do some experiments with several teachers from different backgrounds in order to see: how teachers respond to this; if there is any interest in offering mobile exams; how easy or difficult it is for a teacher to create an exam, etc. Finally, we want to add more different types of items into the Test Editor and the test engines, such as matching-pairs questions, ordering questions, open-ended questions and some others.

REFERENCES

- [1] P. Brusilovsky and P. Miller, Web-based testing for Distance Education. Proceeding of the World Conference of WWW and Internet, Orlando, Florida, 1998, 149–154.
- [2] D. Stephens, J. Bull, and W. Wade, Computer-assisted assessment: Suggested guidelines for an institutional strategy, *Assess Eval High Educ* 23 (1998), 283–294.
- [3] R. B. Clariana and P. E. Wallace, Paper-based versus computer-based assessment: Key factors associated with the test mode effect, *Br J Educ Technol* 33 (2002), 595–904.
- [4] B. J. Halpert, Authentication interface evaluation and design for mobile devices, Proceeding of the 2nd annual conference on Information security curriculum development, Georgia, 2005, 112–117.
- [5] T. Kinshuk, Adaptive mobile learning technologies. <http://www.whirligig.com.au/globaleducator/articles/Kinshuk2003.pdf>.
- [6] U. Farooq, W. Shafer, M. B. Rosson, and J. M. Carroll, M-education: Bridging the gap of mobile and desktop computing, Proceeding IEEE International workshop on wireless and mobile technologies in education, Sweden, 2002, 91–94.
- [7] F. Paterno and C. Santoro, A unified method for designing interactive systems adaptable to mobile and stationary platforms, *Interact Comput* 15 (2003), 349–366.
- [8] E. Basaeed, J. Berri, M. Jamal, and R. Benlamri, Learner-centric context-aware mobile learning, *IEEE Multidiscip Eng Educ Mag* 2 (2007), 30–33.
- [9] G. V. Davidson-Shivers and K. L. Rasmussen, Web-based learning desing, implementation, and evaluation, Prentice Hall, Allyn & Bacon, 2006.
- [10] R. Conejo, E. Guzmán, E. Millán, M. Trella, J. L. Pérez-De-La-Cruz, and A. Ríos, SIETTE: A web-based tool for adaptive testing, *Int J Artif Intell Educ* 14 (2004), 1–33.
- [11] H. Wainer, Computerized adaptive testing: A premier, New Jersey, Lawrence Erlbaum Associates, 2000.

- [12] Microcat, <http://www.assess.com/>.
- [13] FastTEST, <http://www.assess.com/>.
- [14] QuestionMark, <http://www.questionmark.com/>.
- [15] Hot Potatoes, <http://hotpot.uvic.ca/>.
- [16] Webassessor, <http://www.webassessor.com/>.
- [17] A. Boyle, D. Edwards, D. Mackenzie, B. Mills, D. O'Hare, E. Morris, C. Paul, H. Wilkins, and D. Williams, Developments in online assessment—Experiences and evaluation of using TRIADS and its potential for assessment in electrical engineering, *Int J Electr Eng Educ* 37 (2000), 74–85.
- [18] M. Barra, G. Palmieri, S. Napolitano, V. Scarano, and L. Zitarosa, Adaptive testing by Test++, proceeding of the international conference on adaptive hypermedia and adaptive web-based systems, Trento, Italy, 2000, 264–267.
- [19] P. Brusilovsky, S. Sosnovsky, and O. Shcherbinina, QuizGuide: Increasing the educational value of individualized self-assessment quizzes with adaptive navigation support; Proceeding of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education, Chesapeake, VA, 2004, 1806–1813.
- [20] MoodleMobile, <http://jamiap.org/mobile/>.
- [21] M. C. Mayorga and A. Fernández, Learning, tools for java enabled phones: An Application to Actuarial Studies, Proceeding of the International Conference Mobile Learn, London, 2003, 95–98.
- [22] E. Triantafyllou, E. Georgiadou, and A. Economides, CAT-MD: Computer adaptive test on mobile devices, Proceeding of Current Developments in; Technology-Assisted Education, Sevilla, Spain, 2006, 773–777.
- [23] H. Susono and T. Shimomura, Using mobile phones and QR codes for formative class assessment, Proceeding of Current Developments in; Technology-Assisted Education, Sevilla, Spain, 2006, 1006–1009.
- [24] Mobile EMT-B Quiz, <http://www.emszone.com/mobilequiz/>.
- [25] Go test go, <http://www.gotestgo.com/>.
- [26] Mobile quiz, <http://www.gi-de.com/>.
- [27] eInstrucion, <http://www.einstruction.com/>.
- [28] Quizzler, <http://www.quizzlerpro.com/>.
- [29] T. Goh, T. Kinshuk, and T. Lin, Developing an adaptive mobile learning system, Proceeding of the International Conference on Computers in Education, Nevada, USA, 2003, 1062–1065.
- [30] P. Lalos, S. Retalis, and Y. Psaromiligkos, Creating personalised quizzes both to the learner and to the access device characteristics: The Case of CosyQTI, Proceeding of Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia, Amsterdam, 2005, 1–7.
- [31] S. Bull, Y. Cui, A. T. McEvoy, E. Reid, and W. Yang, Roles for mobile learner models. Proceeding IEEE International Workshop on Wireless and Mobile Technologies in Education, Taiwan, 2004, 124–128.
- [32] P. De Bra, A. Aerts, B. Berden, B. De Lange, B. Rousseau, T. Santic, D. Smits, and N. Stash, AHA! The Adaptive Hypermedia Architecture, Proceeding of the ACM Hypertext Conference, UK, 2003, 81–84.
- [33] T. H. Wang, K. H. Wang, W. L. Wang, S. C. Huang, and S. Y. Chen, Web-based Assessment and Test Analyses (WATA) system: Development and evaluation, *J Comput Assist Learn* 20 (2004), 59–71.
- [34] D. Bacon, IMS Question and Test Interoperability, *MSOR Connections* 3 (2003), 44–45.
- [35] W. H. Rice, Moodle E-learning course development. A complete guide to successful learning using Moodle, Packt publishing, Birmingham, United Kingdom, 2006.
- [36] Web-CT, <http://www.webct.com/>.
- [37] W. Stuerzlinger, O. Chapuis, D. Phillips, and N. Roussel, User Interface Façades: Towards Fully Adaptable User Interfaces.
- [38] S. R. Young and L. S. Tonya, Reliability and validity of the mobile phone usability questionnaire (MPUG), *J Usability Stud* 2 (2006), 39–53.

BIOGRAPHIES



Dr. Cristóbal Romero is an assistant professor in the Computer Science Department of the University of Córdoba, Spain. He received his PhD in Computer Science from the University of Granada in 2003. His research interests lie in artificial intelligence and data mining in education.



Dr. Sebastián Ventura is an associate professor in the Computer Science Department of the University of Córdoba, Spain. He received his PhD in the Sciences from the University of Córdoba in 1996. His research interests lie in soft-computing and its applications.



Dr. Paul De Bra is a professor in the Computer Science Department of Eindhoven University of Technology in the Netherlands. He received his PhD in Computer Science from the University of Antwerp in 1987. His research interests lie in adaptive hypermedia systems and web-based information systems.