A Trusted Assessment Environment in a Network-booted OS

Giorgio Biacchi, Philip Grew, and Elena Pagani
Department of Information Science and Communication
Università degli Studi di Milano, Italy
email: {biacchi,grew,pagani}@dico.unimi.it

ABSTRACT

This paper describes how a laboratory where students usually access the LAN using personal accounts can be configured during examinations. Test questions are automatically corrected by a learning-management system, which is used to authenticate students (who are further identified by physical presence). During assessment conditions, students must have access only to the URL where the test is located. Access must be denied not only to other websites but also to any notes or materials the students may have stored in their personal space on the LAN. The authors previously implemented such access policies through Microsoft Windows® user-profiling but have now achieved promising results by booting into RAM a customized GNU/Linux operating system over the network.

KEYWORDS

testing and assessment issues in web-based education; computer-assisted assessment; platform integration; learning management system; innovative web-based teaching and learning technologies; authentication.

1 Introduction

There can be little doubt that computer-assisted assessment (CAA) offers numerous advantages over paper-based testing. What instructors' eyes would not light up at the thought of automated correction of multiple-choice questions—or even simply no longer having to decipher students' nervous handwriting on open-answer essay questions? Fill-in-the-blanks questions and matching exercises are also well suited to take advantage of CAA systems. At a further level of sophistication, natural-language processing (NLP) promises to allow at least partially automated assessment of written or spoken open answers. However, as has been noted elsewhere [1], many instructors show reticence to “tackle" the twin “hurdles" of “rigorous student authentication and a non-trusted environment," thus preferring to continue using paper-based exams.

Browser-accessible platforms used for testing and assessment under controlled conditions, whether designed specifically for testing or consisting of features incorporated into the same platform used for distance learning or online classwork or laboratory projects, require that students seated at a terminal under the vigilant eye of a proctor log on so as to be authenticated. In theory this would seem a simple matter of loading the exam questions onto the learning-management system (LMS) where the students have accounts and setting up the computer lab so that the terminals can access only the URL that corresponds to the LMS. How hard can it be to put test questions on your LMS and reserve the use of a computer room for examination day, after all?

In practice, myriad minor issues, from the distribution of passwords that students must be prevented from exchanging to blocking access to local system features—just imagine a math test during which students discover that typing <ctrl+alt+E> will open a *run* window into which they can type “calc.exe"—make arranging a computer-based test a daunting experience. A representative overview of such issues is given in [2], to which we must add system reliability, an issue rightly emphasized by Cerny and Wrubleski [1]. Indeed, when a server goes down during a test, the socio-pedagogical phenomenon can be likened to a natural disaster. The RAM-disk solution described herein also increases reliability because, once up and running, each student computer acts as a hobbled network terminal with only a browsing function but is independent of any network profile or proxy server.

Our solution to the hurdle of authentication is described in the following section on LAN integration, while two different approaches to creating a trusted assessment environment, with the LAN profile and the GNU/Linux RAM disk, respectively, are described in the subsequent two sections.

2 Integrating the LMS with the LAN

Over the last several years, we have designed, implemented, and optimized an LMS that runs on Apache, MySQL, and PHP—known as *JLI!-Just Learn It!* [3]—both for remote access to course materials and, especially, for bricks-and-mortar testing. Much of the experimentation has involved English-as-a-foreign-language (EFL) classes. A required first-year course, EFL had large enough enrollments to provide the economies of scale that would make automated assessment pay off. The same platform is also widely used for other subjects taught in our department, primarily for assessment purposes.

Thousands of students have now been examined using JLI! [4], which by itself is sufficient for the task, i.e. through authentication on the LMS server. Prospective stu-

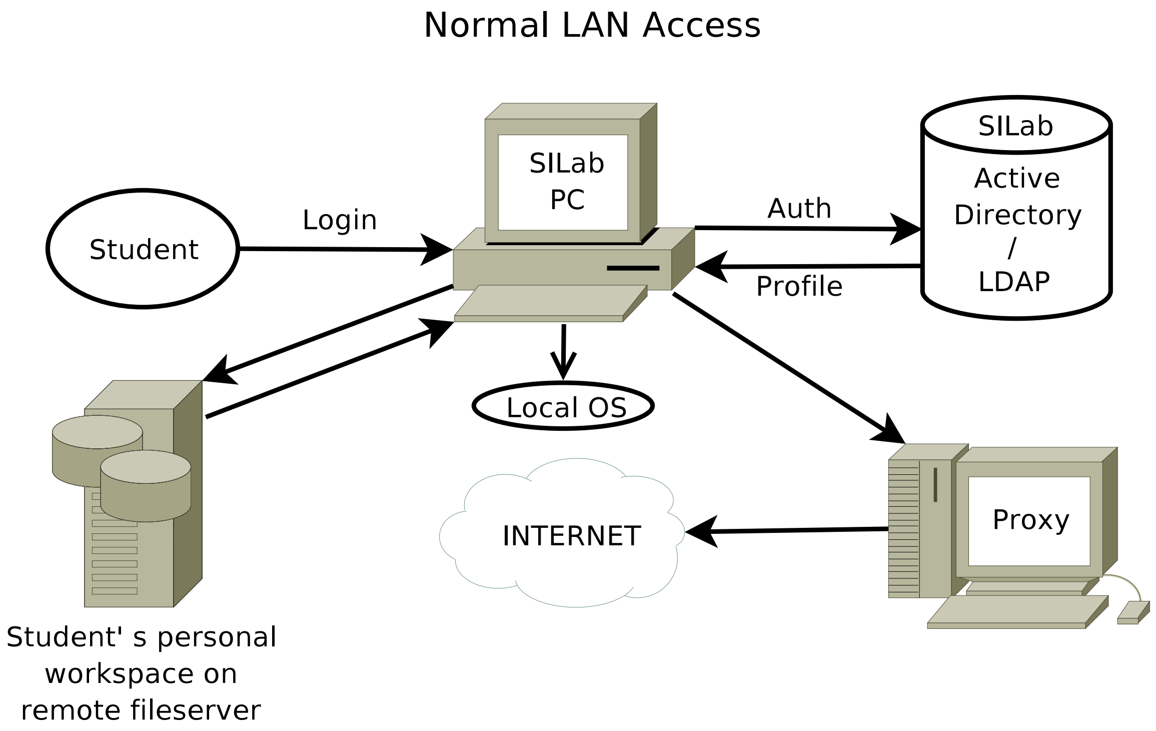

Figure 1. Everyday student use of SILab

dents who have no registration can, for example, be given user IDs and passwords when they show up at the examination on the day of a placement test. Although scripts to generate random passwords, batch commands to create new accounts, and print-merge features to personalize handouts ease much of this process, distributing a few hundred user codes is nevertheless a logistic nightmare. In essence, this hurdle presents itself in such a way because being able to integrate assessment is a very attractive additional feature of the LMS, rather than its core business.

This authentication method works for our exams because our assessment events take place in real time and in physical presence. Students are required to show identification on entering the test center and cannot exchange passwords because each user's name is displayed in very large type on her or his own screen. They keep their photo ID on the desk next to the keyboard, allowing monitors and proctors to easily make sure the student's actual face corresponds to the ID photo and the screen name matches the name on the ID. Except in the case of entry tests, the overwhelming majority of students are known to at least one of the lab assistants, monitors or proctors, in any case.

To overcome the hurdle of password distribution, we rely on the authentication-layer database integration detailed in [5], i.e. students are authenticated for exams using their own real accounts at the Science of Information Laboratory known as *SILab*. Because all assessment events except entry tests actually involve only registered students, we deemed it logical to apply students' existing SILab LAN accounts to LMS authentication. Every computer-science student can automatically open a personal account at SILab, which gives access to a variety of online resources under three operating systems. This LAN account, shown in Figure 1, is used for real programming tasks and is not simply a user login with netcafé-style privileges, though it also provides client functionalities such as email, web access, and printing.

When students log on to JLI!, the LMS authenticates them using the Lightweight Directory Access Protocol (LDAP) [6] to verify their Microsoft Active Directory accounts at SILab (regardless of the student's location). By treating authentication as the central service, we have had a relatively easy time of getting various functions, including the LMS used as an assessment platform, to blend into a larger system. This blended system was not set up merely to provide centralized authentication for the LMS but one of its advantages is that it enables the LMS to outsource authentication to a repository that interfaces with several different databases. As a matter of fact, because JLI! is based on open-integration principles, it can easily obtain certain services from other parallel systems. For example, rather than opening a new forum on the JLI! server, a teacher may simply import forum services from the FirstClass [7] server run by our department's Community Informatics Lab [8], which also hosts the local civic network, Rete Civica di Milano or “RCM" [9]. Moreover, those portions of student records that instructors are legally allowed to view are imported into JLI! from a university-administration database, as described in [5]. Integration achieved at the authentication layer allows various platforms and components to continue independent operation. In our case, such a solution proved more flexible than attempting to export authentication from the underlying LMS as was done by Rizzotti and Burkhart [10].

CAA literature tends to treat student authentication as a major issue [11] but our experience has shown that the problems are not significantly different from the authentication issues that arise during paper-based testing. If anything, CAA allows strict timing, randomization, and other features that enhance authentication [12]. The ultimate effectiveness of the authentication method we are relying on—ID checking—clearly depends on a human component, i.e. the proctor, as an integral part of the authentication chain. Although further automation is certainly technologically possible, for example through RFID [13] or bar codes [14], adding the human touch to look at ID mugshots has not proved burdensome in SILab, which seats no more than 250 at a time. The number of invigilators required is unaffected by this small additional task.

3 Locking down the lab

The second hurdle, securing the environment, has proven somewhat more challenging than the authentication hurdle. One reason is the degree of technical skill required to properly lock down the examination classroom. For small tests of less than fifty students conducted away from SILab in generic campus computer classrooms, we have contented ourselves with limiting access for the entire classroom to the JLI! server. Such netcafé-style configurations may appear to limit access but actually allow a clever student to circumvent the lockdown. For example, on one occasion in a computer room configured with crippled-down Microsoft Internet Explorer browsers, some students discovered they could open an additional browser window by using the *About* menu. (Security holes like this need not be serious if they are caught by the invigilator's eye, which is always the ultimate backup system and tends to quickly respond to any window with colors unlike those of JLI!, even from across the room.) Netcafé configurations may also prevent students from using MP3 files, which are needed for listening passages on EFL exams. The personnel in charge of such study halls may not always have the time or skill required to perfect the needed mix of limitations and privileges.

For these reasons, we have generally preferred to hold JLI!-based exams at SILab, which is staffed by networking experts. All outbound traffic from this lab goes through a proxy server anyway, for security and caching purposes. Students normally log on to SILab terminals using their accounts on this LAN, where user profiles are stored on the Microsoft Active Directory server and accessed from GNU/Linux or Macintosh terminals, as well. For assessment events, our first approach to lockdown consisted of requiring that students be logged on to SILab terminals as a single fake user whose profile had been customized for examination purposes. The fake user exits the LAN via a different proxy, thus allowing a teacher (even of very limited computer literacy) to configure the JLI! server to accept logins only from the fake user's IP class, which corresponds to an array of SILab PCs. There is no drawback to having all students log onto the LAN as a single fake user because they are uniquely identified by their real student accounts at login to the JLI! server (which authenticates them via LDAP). This architecture is shown in Figure 2 and the reader is referred to [5] for further discussion. While this system can be made to work (and indeed was used for a few years), it has a number of drawbacks due to reliance on Windows profiling and students' access to the local disk, including local applications. Because the software on these machines is very frequently upgraded, the lockdown achieved through Windows profiling proved to be high-maintenance and labor-intensive.

4 A different approach

The new lockdown method being phased in at SILab uses an optimized version of the Gentoo distribution of GNU/Linux. This solution was readily available to us because of a number of existing services that were already in place. Note should be taken of the following resource and service requirements:

- PXE;
- DHCP;
- TFTP;
- Gentoo image for network boot;
- groups of terminals with nearly identical hardware.

The preboot execution environment or PXE [15] was already in use at SILab for distributing software and system upgrades by cloning terminals with Symantec Ghost [16]. This routine is run very often during wee hours.

Each SILab terminal is assigned the same IP address at each boot by the DHCP server on the basis of its MAC address. These addresses are thus not truly “dynamic" but the decision to use DHCP was dictated by the need to clone all the PCs with an identical image containing no static IP address. Student terminals on the SILab LAN have public IP addresses that do not go through a NAT. In assessment configuration, the DHCP server is instructed to send boot requests from certain IP addresses to the trivial file-transfer protocol (TFTP) server. The TFTP server then uses PXELinux [17] to serve those PCs the handcrafted Gentoo netboot image, which is loaded into RAM on the local machines and booted up. This allows a sector of the laboratory to be virtually partitioned off for students taking the test. These students are authenticated on JLI!, as before. However, in this case, the only remaining connection to the SILab LAN or to any SILab storage is indirectly through JLI!, which opens an LDAP session. This architecture is shown in Figure 2.

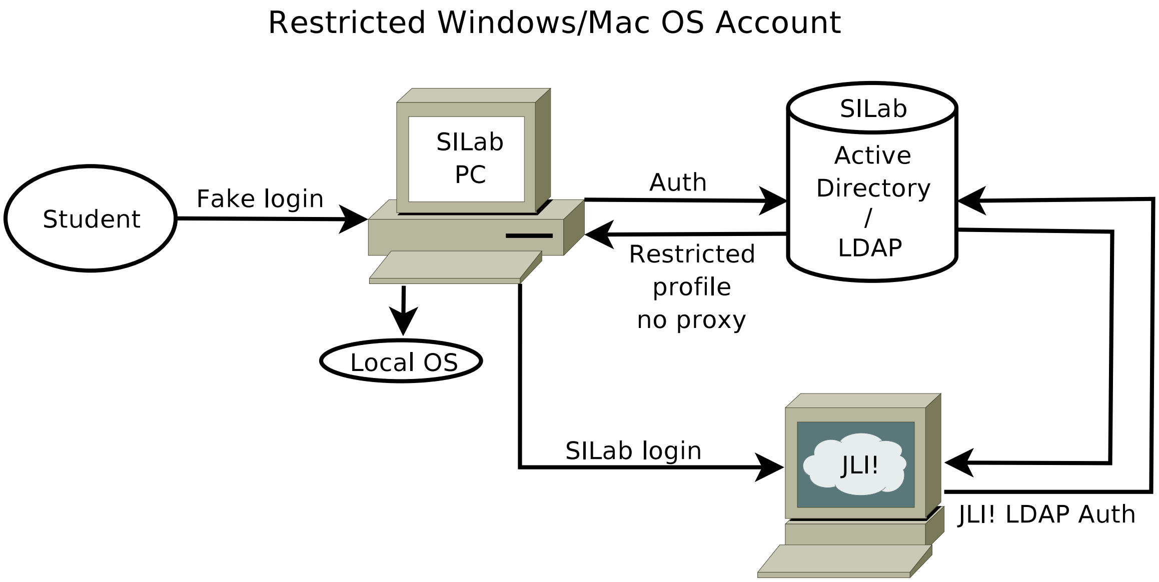

Figure 2. Profile-driven assessment configuration

Preparing the system image: The optimized RAM-disk Gentoo system booted by SILab PCs, which thus turn into netbooted PCs, consists of a stripped-down OS without any applications other than those needed. Many of the daemons and services one would expect to find on a Unix system are absent. The system technically has a single unprivileged user, though this is unseen by the student taking the exam because that user is automatically logged on. The user has no disk access, no LAN access, and no access to removable devices. (It is worth noting that access to USB drives may not be as easy to spot through physical-presence monitoring as other security violations.)

The RAM-disk Gentoo system was compiled on one of the identical SILab PCs, vetted, and then prepared as an image. It allows use of a browser and, at least for an EFL exam, of the application that plays MP3 and/or .wav files, with which these formats are associated by the browser and OS. The key to creating the trusted environment for use of the browser, in our case Mozilla Firefox, is the OS kernel's support for iptables, which allows only one IP address (that of the LMS) to be accessed. The use of iptables obviates the need to cripple the Firefox feature that allows users to choose their proxy server, which had been a problem under Windows. It goes without saying that iptables offers ease of configuration, as well as reliability and security.

Multimedia integration: The EFL tests involve a listening component, which typically consists of a recorded audio attachment in MP3 format, although for very short audio segments .wav files have been used. The latter have the advantage of playing directly in the browser, thus working in any computer room. Many computer rooms on campus have Windows systems that have been crippled down so as not to support the MP3 format. At SILab there is theoretically no problem using this format; yet in reality it has been a source of seriatim headaches caused by applications competing for association with the MP3 format.

On Windows systems, in Microsoft Internet Explorer (the browser lamentably still preferred by many students) right-clicking on the embedded MP3 content plays the audio as a stream, preventing students from pausing or rewinding. Over the years we have experimented with setting up audio playback in various different applications: Nullsoft Winamp, RealPlayer, QuickTime, and Windows Media Player. None has proven ergonomic. Which application will open MP3 files by default depends on the association stored in the fake user's Windows profile. None of the fake user's profile preferences are saved, so any application used to play MP3s during the assessment event will be launched on a first-time basis. On first run, all these applications entail some procedure, such as Windows Media Player's license agreement. The extra clicks needed on first run drastically complicate the soundcheck students are led through at during exam instructions. In addition, these procedures raise the thorny deontological issue of why students being examined at a public university should be forced to become customers, albeit virtually and anonymously, of a private software manufacturer, a creeping ethical issue in much of the public sector, especially in Europe.

By setting up the Gentoo RAM-disk system to silently launch XMMS when the embedded MP3 audio is called up, all these audio troubles have been resolved. In addition, the OS sets up default volumes in hardware and software, obviating much of the adjustment previously needed during the exam-initial soundcheck.

Unresolved issues: The assessment OS image does not contain applications to support all the functionalities avail-

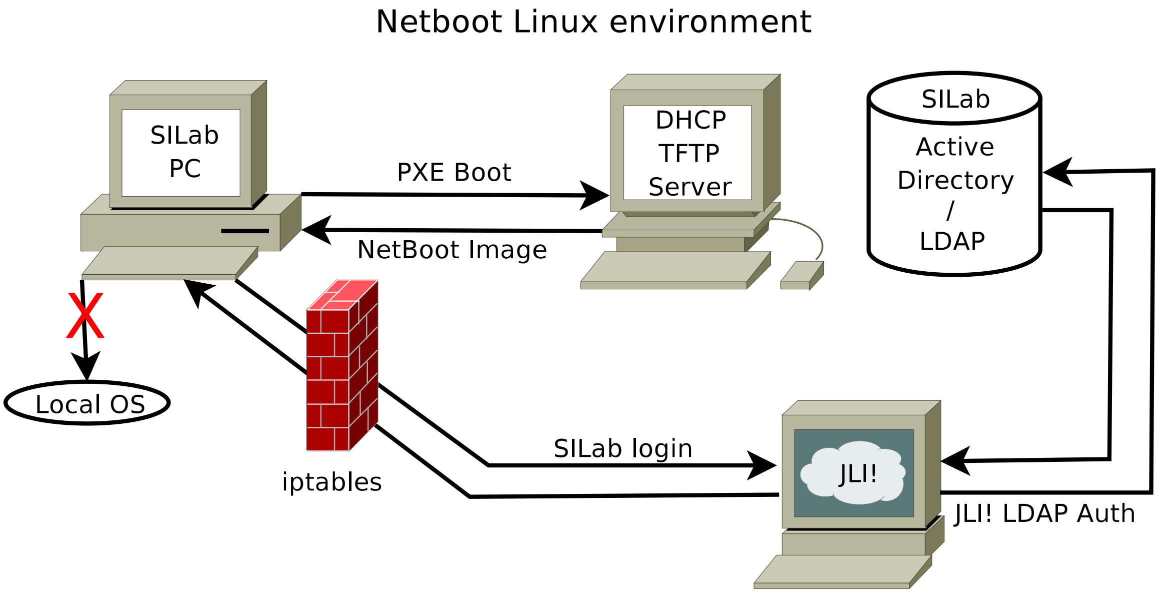

Figure 3. RAM-disk assessment configuration

able on the LMS. For example, an examination requiring students to demonstrate the ability to produce a spreadsheet elaboration of a certain concept would require that OpenOffice be installed in the image. This would likely bloat the image beyond the capacity of currently installed local RAM (one gigabyte). Fortunately, in practice, many of the more sophisticated functionalities JLI! affords, such as embedded video, are rarely used during assessment.

However, other subjects taught in our department do rely on specific funtionalities that are not yet provided for in the Gentoo RAM-disk. For example, the Computer Networking exam mandates that each student write a client program to be compiled in Java and made to interact with a server on the hardware platform that hosts the LMS. A requirement of this sort would necessitate preparing a RAM-disk image that contains a Java development kit and perhaps a text editor other than VI. Further investigation will be required to determine whether we are ever able to have a single image optimized for all the subjects using CAA in SILab. However, subjects other than EFL, because they do not have to deal with audio attachments, have so far had less incentive to shift over to the RAM-disk lockdown method. In any event, it is not a huge challenge to start from the existing image and customize it in different directions for application to various kinds subject matter. The hard part was getting the image to work initially.

The GNU/Linux netboot image we are currently using weighs 191 Mb compressed and loads onto a 640 Mb RAM disk (with enough filesystem space available for downloaded exam materials). There is no doubt that further optimization of the image is possible and will reduce network load at boot time while improving performance during the boot operation.

5 Discussion and concluding remarks

Problems arose with the use of the local Microsoft Windows OS not only because of license issues but also because the system has a wealth of features that cannot normally be disabled without acrobatics worthy of a true computer wizard. After a policy has been implemented, extensive vetting is required to determine that all those components that might be subject to abuse have been effectively disabled.

Once the netboot image had actually been made to work, transition to this lockdown method proved less painful than expected. Greater student resistance was encountered when examinees were obliged to use a specific browser (the near-sighted bemoaning Internet Explorer's lack of text zoom and others lamenting the unfamiliarity of the Firefox or Safari interface. The authors believe that, because students come into the lab and are seated at a terminal with an running browser under the netboot approach, many do not really much care exactly what is running their interaction with the LMS. More significant is the fact that many students are already familiar with JLI! from its use to distribute exercises and collect coursework.

CAA offers a refreshing break from other assessment methods and proves to be a memorable experience for the students. This contributes to the sense of community that fosters class cohesiveness even throughout a large group of students. Nevertheless, there is no denying that our current methods cry out for further automation and improvement. For example, following the assessment event, the instructor must export the data from the exam (as a comma-separated values file) and import it into a spreadsheet application in order to implement weighted automatic scoring or to merge results with assessment activities based on paper quizzes or oral exams. This is part of further development work needed on the LMS itself, as is a dedicated testing mode in JLI!, so as to obviate the need to obscure—on a module-by-module basis—notes or courseware that students are not allowed to access during the exam.

In addition to potential automation of certain segments of the authentication-control process, such as generating lists of attendees via RFID or the like, other enhancements we would like to implement at our assessment events include automated pronunciation testing by having students' spoken output fed into a speech-to-text synthesizer for comparison to text strings programmed into the answer key. Only a trusted environment that guarantees reliability allows further experimentation of this kind. Confidence in the trusted environment, on the part of both proctors and examinees, is a prerequisite for a successful CAA event.

Acknowledgments

The authors wish to thank JLI! developer Ivan Longhi of the Fondazione Rete Civica di Milano, and SILab administrator Giulio Casella of the Università degli Studi di Milano, who have patiently collaborated in analyzing system usability and probing for security weaknesses.

References

[1] Jan Cerny, Marc Wrubleski, "Creating a Trusted Environment in which to Perform Computer-assisted Assessment," *Proc. IASTED International Conference on Web-Based Education (WBE'06)*, Puerto Vallarta, Mexico, 2006, 47-50.

[2] Mike Joy, "Automating the process of skills-based assessment" in Sylvi Alexander (ed.), *Effective Learning and Teaching in Computing (Effective Learning and Teaching in Higher Education)*, RoutledgeFalmer, 2004, 76-85.

[3] Ivan Longhi, Philip Grew, *JLI!-Just Learn It!*, http://sourceforge.net/projects/jli.

[4] Philip Grew, Ivan Longhi, Fiorella De Cindio , Laura A. Ripamonti, "Applying an LMS to Large Language Classes," *Proc. IASTED International Conference on Web-Based Education (WBE'04)*, Innsbruck, Austria, 2004, 480-484.

[5] Philip Grew, Ivan Longhi, Elena Pagani, "Functional Architecture of a Web-based Distributed System for University Curricula Support," *Proc. IASTED International Conference on Web-Based Education (WBE'05)*, Grindelwald, Switzerland, 2005, 332-337.

[6] Kurt D. Zeilenga, *RFC4510* "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map," The Internet Society, 2006, http://tools.ietf.org/html/rfc4510.

[7] Open Text Corporation, "FirstClass Communications Platform," http://www.firstclass.com/AboutFC/.

[8] Community Informatics Lab, "Rete Civica di Milano," http://www.lic.dico.unimi.it/.

[9] Rete Civica di Milano, http://www.retecivica.milano.it/.

[10] Sven Rizzotti, Helmar Burkhart, "Web-based Test and Assessment System: Design Principles and Case Study," *Proc. IASTED International Conference on Web-Based Education (WBE'06)*, Puerto Vallarta, Mexico, 2006, 37-42.

[11] Joanna Bull, Colleen McKenna, *Blueprint for Computer-Assisted Assessment*, RoutledgeFalmer, 2004.

[12] Joanna Bull, Grainne Conole, H Davis, Su White, Myles Danson, Niall Sclater, "Rethinking Assessment Through Learning Technologies," *Proc. Australasian Society for Computers in Learning in Tertiary Education, 2002 Confernece (ASCILITE'02)*, UNITEC, Auckland, New Zealand, 2002.

[13] Philip Grew, Elena Pagani, "Towards a Wireless Architecture for Mobile Ubiquitous E-Learning," *Proc. Intl. Workshop on Learning Communities in the Era of Ubiquitous Computing*, Milan, Italy, 2005, 20-29.

[14] Nagaoka, K., "A Response Analyzer System Utilizing Mobile Phones" *Proc. 4th IASTED Intl. Conf. on Web-Based Education (WBE'05)*, Grindelwald, Switzerland, 2005, 579-584.

[15] Intel Corporation, *Preboot Execution Environment (PXE) Specification*, Intel, 1999, http://www.pix.net/software/pxeboot/archive/pxespec.pdf.

[16] Symantec Corporation, *Symantec Ghost Solution Suite*, http://www.symantec.com/

[17] H. Peter Anvin, *SYSLINUX*, http://syslinux.zytor.com/pxe.php