Mobile Virtual Laboratory: Learning Digital Design

Vlado Glavinić  
Faculty of Electrical Engineering and Computing  
University of Zagreb  
Unska 3,  
10000 Zagreb, Croatia  
vlado.glavinic@fer.hr

Mihael Kukec  
College of Applied Sciences  
Hallerova aleja 5,  
42000 Varaždin, Croatia  
mihael.kukec@vels.hr

Sandi Ljubić  
Faculty of Engineering  
University of Rijeka  
Vukovarska 58,  
51000 Rijeka, Croatia  
sandi.ljubic@riteh.hr

Abstract. An important part of learning digital design is the acquisition of knowledge through working experience in a laboratory environment, what will corroborate theoretical knowledge exposed elsewhere (e.g. in lectures, through written exercise solving etc.). This paper describes the foundation of a virtual laboratory available to every student at any moment and regardless of her/his whereabouts, following the concepts of universal access. We discuss the main issues in running a digital design lab, the essentials of contemporary mobile technology and the possible solutions to merge the two with the final objective being the development of a multiplatform system able to run both on (desktop) PCs and on mobile devices (terminals). We also discuss issues in the supporting HCI, especially in making the interaction as simple as possible.

Keywords. adaptable user interfaces, m-learning, mobile HCI, universal access, virtual laboratory

1. Introduction

Rapid development of new mobile technologies, communication-information infrastructure, and of widely-accepted e-learning systems has made it possible to introduce a new paradigm in the field of distance learning, which is denoted mobile learning or m-learning. Generally speaking, m-learning is the intersection of online learning and mobile computing, which promises the access to applications supporting learning anywhere and anytime [1], implementing the concepts of universal access [2]. The definition of m-learning is to learn knowledge ubiquitously and movably [3].

As with all other new paradigms m-learning has its own advantages and deficiencies. New and fast developing mobile computing devices (mobile phones, pocket PCs, Tablet PCs, and PDAs) and mobile technology provide people with “wearable” computing ability to conveniently participate in distance learning environments [4]. Advantages of m-learning include mobility itself, better time-spending efficiency and more opportunities for peer collaboration. Furthermore, with the use of m-learning, distance education can encompass students with different levels of knowledge, capabilities and learning style categories, as well as anyone who requires mobile computer solutions that other devices cannot provide at the given moment or location. On the other hand, mobile learning disadvantages arise as a result of its general weaknesses, from psychological, pedagogical and technical viewpoints (as described in [5]). This especially applies to the most-frequently used mobile devices – mobile phones. General restrictive factors of these devices are small displays and their low resolutions, input limitations, limitations in accessing the Internet and the lack of standardization and compatibility.

In order to make an effective use of an m-learning system, all available advantages and (we hope) temporary disadvantages are to be considered, along with the type and features of the application to be provided by m-learning means to the users. In this paper, we discuss a specific model of m-learning application, which is to support a mobile virtual laboratory for teaching digital design. The paper is structured as follows: Section 2 explains the motivation to develop such a laboratory, Section 3 shortly outlines available mobile devices and pertinent technologies, Section 4 delves into specific requirements for the mobile virtual laboratory model, and Section 5 discusses the relevant HCI issues.
2. Motivation

The idea to build a mobile virtual laboratory comes as the result of several years of observation of students in several forms of laboratory training in digital design. Traditional training takes place in a specially equipped laboratory where students model and implement digital circuits using protoboards and integrated circuits, the correctness of their design being verified on standard measuring instruments, as shown in Fig. 1. Due to both a limited quantity of (relatively expensive) equipment and a usually large number of students, this form of laboratory training becomes increasingly ineffective and time-consuming.

On the other hand, digital circuits modeling and design can be performed on (desktop) PCs by using appropriate development and simulation packages. Aside providing students with a new dimension in design, since it enforces the use of a suitable hardware description language (e.g. VHDL [6], see Fig. 2), the number of possible student assignments/projects can be increased, imagination being the only limit. Additionally, problems linked to failure of expensive measurement instruments are not present any more. However, let not be missed the problem of administration and coordination in such a computer laboratory, possibly involving quite a number of student groups.

The issues referred to above influenced the modeling of a system that would enable distance-based laboratory training using either wireline interconnected computers (e-Lab), or wireless interconnected mobile devices (m-Lab). This combined functionality actually describes a complete model of a virtual laboratory (Fig. 3). This paper concentrates on the mobile aspect of the virtual laboratory, which we expect to become a very useful m-learning service.

3. Mobile devices and available technologies

The idea of a mobile laboratory for learning and training digital design is supported by several factors, the first being the rapid growth of information and communication technology [7]. Mobile and handheld devices (PDAs and mobile phones of various types) are becoming both more powerful and versatile, having the ability to connect to the Internet and perform high data rate transfers. These characteristics, combined with decreasing prices, have made those devices very appealing and in fact quite common among the student population. On the other hand, the software running on these devices is predominantly designed to support the business world, with typical applications including time management, and communication and productivity tools [8]. Actually there is almost no software on mobile devices that would support any kind of learning or even curriculum delivery, although some of them run applications for reading e-books (which are mostly third-party applications). However, there already exist studies in m-learning (e.g. [9]), as well as working m-learning applications (e.g. [10]).
From the application development perspective mobile phones can generally be divided into two groups, the important difference between them being their ability to support application development (see Fig. 4). The first group consists of devices controlled by proprietary operating systems (OSs), which are usually model specific (e.g. Nokia mobile phones running on the "Series 40" platform [11]), while the other of smart-/mobile phones (where a distinct line between these two device types is somewhat blurred), which are usually controlled by more complex OSs (e.g. Symbian, Windows Mobile, Palm OS, or Blackberry OS).

Figure 4. Handheld devices

PDA devices generally have more powerful processors and larger memories hence no problem arises with applications to be shared with mobile phones. J2ME support is implemented in the most common PDA operating systems: PalmOS and Windows Mobile. PalmOS supports J2ME through IBM WebSphere Everyplace Micro Environment Java J2ME runtime, while most of the current Windows Mobile based PDAs have third-party Java Virtual Machines pre-installed [14]. There seems to be quite a homogeneous situation regarding J2ME platform support in mobile phones since it usually is already installed in the OS. This is however not the case with PDAs where sometimes the user has to install the support on his own and in some cases pay extra for it. Such a situation is going to be improved with packages like phoneME – an open-source project trying to further expand the usage of Java™ Platform, Micro Edition (Java ME platform) technology in the mobile handset market [15].

Figure 5. Sample J2ME Stack

When speaking of J2ME support, we assume that the J2ME software stack, see Fig. 5, is implemented in the mobile device. The lower
layer CLDC (Connected, Limited Device Configuration) defines a configuration which determines all that is needed in order to have basic Java Technology support in a compliant device, while the higher one MIDP (Mobile Information Device Profile) defines the profiles which provide industry-specific or specialized APIs on top of the underlying configuration. CLDC and MIDP together specify the core VM and Java language features plus the wireless and mobile-specific APIs required for mobile devices [16].

Another advantage of the J2ME platform is the simplicity in implementing the model of interactive communication between the mobile device and the distant Internet-connected server. Since all MIDP implementations must have support for HTTP, interactive exchange of data between these two entities can be created as a cooperation of MIDlet and Servlet applications, as shown in Fig. 6. Here MIDlet represents any MIDP application which can be realized on a mobile device while Servlet represents a server application based on the widely-accepted J2SE platform.

4. Requirements for mobile virtual laboratory model

In the remainder of the paper we explore the virtual laboratory model by discussing the solutions available for obtaining a multipurpose system with applications able to run on PCs as well as mobile devices. System tasks are (i) to deliver interactive contents for learning and training, (ii) to guide through the entire learning process (including an evaluation mechanism), and (iii) to administer all users. The delivered learning contents usually consists of multimedia materials which combine texts, pictures and audio files. More complex interactive applications enable more sophisticated tasks, like laboratory training or testing. Delivered content could be shaped as a set of Mobile Interactive Learning Objects (MILO), see [17], of a high abstraction level, hence having a wide area of possible implementation. To work in a specific field it is necessary, and possible, to develop interactive objects that would support the studied materials. A MILO set used in teaching and learning digital design, along with the associated laboratory training, can be further expanded with a tool for modeling and simulating digital circuitry. Such tool should enable the users not only to model and simulate, but also to group simple logic models into more complex ones, with the extra feature of storing them for future usage. Furthermore, the system must make it possible for the author of the objects to create certain sets of assignments and exercises that can further be used to automatically generate assignment instances later on. It should also enable the author to implement an automatic evaluating system for solutions received. According to evaluation results the next action can be set: if the solution is not satisfactory, the system can inform the user and with an adequate help, meaning extra information, gradually guide her/him to the correct solution of the exercise. If on the other hand the solution is completely, or partially, correct, it can further examine the user or it can direct her/him to learn other parts of the curriculum and do other laboratory assignments.
The learning and training cycle for a mobile virtual laboratory is shown in Fig. 7. It illustrates a model of a laboratory system where the support for simulating logic circuitry being modeled on the user's side, is on the mobile device itself. This means that in such a model implementation that particular component must be separately developed, as e.g. a lightweight Java simulator whose executive code must not exceed the memory limits of target devices. As opposed to such a model it is possible to use an already complete solution instead of creating one's own; this solution is regularly too demanding for the memory of handheld devices, and in most cases is not realized with Java technology, hence it is transferred to the server side of the virtual laboratory system, see Fig. 8. The user of the mobile device receives the object describing her/his laboratory assignment, goes through circuitry modeling using the application running on her/his device and eventually forwards the developed model onto the m-service server. The server based simulator is capable of interpreting the received data and returns the generated results to the user application, which in turn visualizes them. In the process, the user also receives information on possible modeling errors. Upon a user's mistake the system can decide whether to re-examine her/him, set easier laboratory assignments, or "arrange" for an additional delivery of learning materials concerning the specific laboratory exercise.

5. HCI issues

Using a learning system in a mobile environment is not effective if obstacles are not overcome in the interaction with the targeted user(s), which results from limited both input and output capabilities of mobile devices. For that purpose all mobile services, including m-learning ones, demand special attention being paid to interaction with the user [19]. Developing all-present learning applications also involves developing an adjustable interface to work with different devices where the application is used [20].

5.1. Device-based adaptable user interface

The final objective being the development of a virtual mobile laboratory to be accessed from a (desktop) PC, a mobile phone and a PDA, it should be noted that all the mentioned devices have their own specific features which are significantly different, what brings us to the issue of creating a unique application having functionally the same or very similar user interface (UI). Section 3 discusses why Java (J2SE and J2ME) is the most suitable technology and the logical choice for development of such an application. There are however technologies enabling development of applications with a unified UI for different types of handheld devices [21, 22], but they only offer APIs for supporting general-purpose widgets exemplified by graphical UI (GUI) elements such as buttons or editable fields. This is not sufficient in our case since the user interface must show more complex graphic elements like symbols of digital circuits and virtual devices. Besides, these technologies are not entirely useful because they are not compatible with the Java technology although they were made to resemble it.

A mobile virtual laboratory application can be divided into three functional blocks: (i) user interface and interaction with the user, (ii) communication between interface and simulation components, and (iii) components for simulation of digital circuitry. As a detailed analysis of communication and simulation components is beyond the scope of this paper, in the following we will concentrate on the user interface and the associated interaction.
In order to present the application interface in a device-independent way, it is necessary to develop a virtualizing layer between the device and the system core, its primary function being to receive user commands across the interface. While doing that, the virtualizing layer must be able to interpret actions originating from all three device categories and relay them into the system core in a unified way. This is because the same action is performed differently on different types of devices, using different interaction styles. E.g. a desktop user will mostly use the mouse and to a lesser extent the keyboard, the user of a mobile phone will use numeric and navigation keys or even micro-joysticks, whereas a PDA user will perform the same actions by using a "soft" keyboard on a touch-sensitive display.

The second function of the virtualizing layer is the presentation of the interface on a specific device display. The system core identifies the user’s actions and, in accordance with their meaning, generates the interface on the canvas of the virtualizing layer, hence enabling it to create the image on the target display. Fig. 9 shows the implementation details of the system core and the virtualizing layer.

![Figure 9. UML class diagram for adaptable user interface](image)

It can be seen that the system core is represented as a UserInterfaceEngine class, whereas the virtualizing layer is represented by the cooperation of this and the VirtualCanvas class. AppletCanvas and MIDLetCanvas represent classes which describe a computer monitor and a mobile phone display, respectively. System operation follows the next steps: a PC application (e.g. an Applet), and a mobile phone/PDA MIDLet will implement no interface functionality (neither in their own classes nor in any directly inherited classes) except the communication with the core, i.e. the UserInterfaceEngine object. This latter object processes users' actions, like pressing a key or moving the mouse, and in accordance with the new situation generates the interface in a uniform way disregarding the type of the access device. The VirtualCanvas object generates the image on the specific access device display. Hence, the virtualizing layer is realized through the cooperation of the core, the UserInterfaceEngine, and the VirtualCanvas objects, which reconciles the differences while generating the image on different devices.

5.2. Towards an intelligent interface

The ultimate objective to be addressed in m-learning is clearly UI automatic adaptation in accordance with both user preferences and level of knowledge.

In order to make system use as simple and efficient as possible, the user interface should actively monitor the user while interacting with the system. This involves installing suitable adaptation algorithms which should operate “in the background” and, based on data thus collected, change the momentary interface look. Applicable algorithms can be (i) recency-based, monitoring the most-frequently used commands during the entire interaction, or (ii) frequency-based, calculating the frequency of usage of certain commands in shorter interactive cycles. E.g., [23] shows how it is possible to realize different types of interface adaptation basing on these algorithms. Furthermore, user interface adaptation need not develop only towards personalization of interaction but also towards generating very simple – minimalistic – user interfaces.

Automatic adaptation already touches upon some intelligent procedures in the system-user interactivity domain by modeling a user behavior profile, cf. [24]. This profile, which describes the preferred way of interaction, can also be used in anticipating future user objectives in specific tasks of her/his work process. Such objectives anticipation allows greater interaction efficiency and speed by introducing system interface "shortcuts", as well as auxiliary information on the interface associated to the current task. Along
with behavior profile modeling, it is also possible to model a user knowledge profile (cf. [25]), containing and updating user information which would help the system to determine the category of her/his both knowledge and skills in certain fields of expertise (e.g. beginner, intermediate, advanced, expert). Based on these information it would be possible for the user interface to make the work easier for each user by automatically adjusting itself during the usage of the system.

6. Conclusion and future work

M-learning systems will replace neither the traditional way of learning nor the already widely-accepted e-learning services. They will however certainly become an additional benefit in the comprehensive process of lifetime learning. In the paper we considered the possibility of performing simple laboratory exercises using mobile devices, what also brings us closer to implementing a full-scale mobile virtual laboratory.

Capabilities of mobile devices, their input and output, are still not at a level where their applications could compete with some standard desktop applications. However, the contemporary development of mobile technologies and the associated information-communication infrastructure do make possible the introduction of m-learning through delivering various multimedia and interactive contents for learning and interaction between mobile users and m-learning servers. Developing a virtual laboratory system which would offer on the one hand e-learning services, usually on desktop computers, but also on the other adequate m-learning services for users of handheld mobile devices, can definitely be considered a significant breakthrough in the field of computer supported education.

This paper proposes a model of such a virtual laboratory, which supports m-learning and training in the field of digital design by (i) delivering interactive multimedia learning contents to mobile devices, (ii) modeling logic circuits with an application on the mobile device, (iii) simulating circuit models on a distant m-learning server or on the mobile device itself, and (iv) visualizing simulation results in accordance with the capabilities of the used mobile device. HCI aspects of the system are also discussed, addressing long-term objectives as well as a possible solution for user interface adaptation in agreement with the type of the access device.

We are continuing our work on the implementation of the model presented by primarily focusing on the development of mobile device applications, where we expect to fully show the advantages of the described mobile HCI. These applications will be tested in various working environments, with the results thus accomplished being the measure of user efficiency and satisfaction analysis.

7. Acknowledgements

This paper describes the results of research being carried out within the project 036-0361994-1995 Universal Middleware Platform for e-Learning Systems funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

8. References


