MOBILE: A MOBILE Instructional Laboratory Environment for Hands-on Information Technology Education

Paul Wagner
wagnerpj@uwec.edu

Chris Johnson
johnch@uwec.edu

University of Wisconsin, Eau Claire
Eau Claire, WI, USA

ABSTRACT
We present MOBILE, a software system providing ad-hoc and hands-on learning environments for information technology education. Using MOBILE, instructors can hold workshops on any available computers, even in restricted lab settings or using participant computers. Features supported by this software include the ability to build and configure a workshop session on any desired information technology topic, creation and management of that session on top of any existing network, distribution and configuration of additional operating systems and applications, and the inclusion of auxiliary content relevant to the session. In this discussion paper, we describe the MOBILE infrastructure and explore an actual workflow for an instructor using it to teach a workshop on computer security. We have successfully piloted the system to conduct four highly-rated workshops at technology education conferences.

Categories and Subject Descriptors
K.3.2 [Computer and Information Science Education]: Computer science education

General Terms
Management, Experimentation

Keywords
hands-on learning, ad-hoc laboratories, workshops

1. INTRODUCTION

Educators face a tremendous question as they plan for workshops and outreaches: what is the best way to present the material? Most would agree that a hands-on, active learning approach engages students and has a greater educational impact. However, in information technology, active learning often means a computer is involved, and the prospect of setting up a lab full of computers is intimidating to many and time-consuming for all. Perhaps the venue is a school in which the lab machines are locked down and whose IT staff is not interested in installing new software. Or perhaps the workshop is at a conference, and the lab is an amalgamation of machines assembled a few hours before the presentation. Maybe participants are bringing their own machines. Whatever the hardware infrastructure may be, creating a successful hands-on learning environment to teach information technology means addressing the problem of unknown and heterogeneous client machines. As a solution, we present MOBILE, a mobile interactive learning environment for information technology education. MOBILE is a software system that homogenizes a diverse information technology learning environment without burdening IT staff or permanently altering the client machines.

This project was born out of the frustration of the authors, who’ve been frequent instructors of and participants in workshops at conferences and schools. More than 50% of the workshops we’ve attended have been plagued with hardware and software issues, including a) limited lab space due to high demand from concurrent conference sessions, b) limited physical access to host site machines, and c) restricted software privileges on host site machines. These issues led to the prototyping of a software infrastructure that runs on top of a diverse set of machines, including participants’ personal computers, without modifying them in any way. This prototype was used to successfully deliver four large-scale and highly-rated computer security workshops at computer science education conferences. MOBILE is a generalization of this prototype, such that it can be used to administer an ad-hoc laboratory for almost any information technology topic.

MOBILE solves the challenges of using machines outside of the instructor’s control by providing a baseline operating system, a customization of Ubuntu Linux, which participants boot into from a portable storage device. Layered on top of this system is a communication utility the instructor uses to dynamically share files with participants. Linux is not suitable for every workshop, and for this reason, the MOBILE system may be populated with virtual machine images preconfigured by the instructor. With a little advanced preparation of the MOBILE system, an instructor may expect uniform software configurations and focus on delivering content instead of providing technical support.

In this paper, we describe the MOBILE project in detail. We discuss in Section 2 relevant related work, in Section 3 its software architecture, in Section 4 our experiences in
successfully using it to conduct several workshops, and in Section 5 our future plans for MOBILE.

2. RELATED WORK

Our development of MOBILE as an instructional tool builds on two main ideas: a) that laboratory-based learning is a valuable component in information technology education and b) that the administrative challenges instructors face in physical laboratories can be overcome. Work related to ours addresses these two points.

2.1 Active learning

Laboratory experiences complement the passive learning that comes from reading and listening to lectures, and no one questions their educational value. Nersessian [20] states that “hands-on experience is at the heart of science learning.” However, it’s not as clear what form the laboratory experience should take. Corter et al. [11] explore the question of whether remote, web-based labs can be as effective as hands-on labs. While the authors find that students can learn as much in remote labs as they can in hands-on labs, students strongly favored hands-on labs, ranking physical presence as the most important factor in effective lab experiences. Participants in the study commented that the inability to ask questions and clarify instructions was a major concern for remote labs.

Kumar [18] investigated the influence of labs on retention in a computer science program. His work contrasts open labs, which consist of exercises students complete on their own time, with closed labs, which are scheduled and structured. Switching to closed labs had no impact on retention in the study, but the assessment indicates that closed labs have a positive impact on student learning, especially in the first part of a course.

These studies, a few among many [24, 6, 19], strongly suggest that hands-on labs serve a crucial role in education. Despite the complications and cost of setting up a physical laboratory, the social and communication benefits are significant.

2.2 Lab Administration

Administering a lab full of machines has become a much simpler task in recent years, thanks largely to better software tools [14]. We discuss several aspects of configuring and maintaining a lab that are relevant to our project.

2.2.1 Efficiency

The improved efficiency of configuring and maintaining a lab has mostly been a result of virtualization. Virtualization decouples the notion of a computer from discrete physical hardware; it is the “logical representation of a computer in software.” [16]. There are many reasons virtualization is becoming increasingly popular, including reduced costs, improved hardware resource utilization, and lower energy consumption through the use of thin clients.

Educators have reported frequently the merits of virtualization in laboratory settings [9, 3, 24]. Van Aardt and Mossom [23] summarize the well-known dangers of running a physical laboratory: high cost, hardware failures, operating system (OS) breakage, and significant instructor time spent reimaging the systems for each new project. After switching to a virtual environment, they reduced the time to reimage their labs to just a few minutes.

The MOBILE project strives for efficiency in a slightly different way. Its intended purpose is to serve presenters offering workshops away from their home environment, such as at other schools or conferences. In such settings, the presenter has little control over the machines, which may not have the necessary software installed to conduct the workshop. MOBILE provides a baseline OS to run on the hardware that is available, turning a heterogeneous lab into a homogeneous one.

We note that while virtualization is a feature directly supported by MOBILE, the MOBILE OS itself runs on a real machine, not a virtual one. Participants boot into it from a portable storage device. We chose this approach in order to have complete software independence from the native machines, on which no guest accounts or virtual machine managers are needed.

2.2.2 Flexibility

Many researchers have reported on ways to assemble labs for special purposes, like teaching networking or computer security [24, 15, 10, 7]. Often such labs have been designed to work under tight budgets [12, 13], and nearly all of them employ virtualization to reduce costs and allow script-driven maintenance. From an instructional perspective, it is important that labs remain general purpose—one lab, virtual or physical, may serve many purposes [3, 9, 4].

MOBILE has its origins in a software infrastructure for conducting workshops on computer security. However, MOBILE is an intentional repurposing and abstraction of this first project and has been designed to meet the needs of any software-driven lab.

Several curricula have been published that support physically portable labs [22, 5]. One of MOBILE’s primary goals is to allow presenters to conduct labs on the road, at other schools or conferences where the presenters may have little control over host machines.

A project similar to ours is CPVM [8]. Butler et al. designed CPVM to provide a completely portable and predictable development environment for their students and for faculty away from their offices. By design, their environment is restricted in size and builds off a processor-emulation layer. Users still need access to host machine software. With MOBILE, we offer a full computing environment that runs directly on the hardware without requiring user accounts.

2.2.3 Sandboxing

To learn, students need to have freedom to break things [13]. One of the features of virtualization most widely reported in recent years is the ability to create sandboxed environments [17, 4]. Users custom-build virtual machines configured for a particular purpose; these self-contained images do not interfere with other images or endanger the user’s native operating system and personal files. MOBILE similarly protects the workshop participant’s machine by imposing an instructor-configured baseline OS atop the client hardware.

3. DESCRIPTION

3.1 Architecture

3.1.1 Overall architecture

MOBILE is designed with two major goals: a) to provide a standard yet flexible hands-on instructional environment
using a heterogeneous set of computers brought by workshop participants or provided by a hosting venue and b) to protect those various systems from any harm. The environment is launched by booting MOBILE from either a system’s hard disk or, more likely, an external storage device. Booting from an external device allows MOBILE to use a system’s processor and memory without reading from or writing to any internal hard drives, thereby protecting the computers from configuration or data modification.

Booting MOBILE directly from USB devices is possible on Windows or Linux systems. With the aid of an auxiliary boot CD, MOBILE can be booted into indirectly on Macintosh systems. The foundation of MOBILE is a modified version of Ubuntu Linux that provides a standard base platform for the instructional environment as well as additional functionality for managing workshops and interacting with participants.

Suppose an instructor wishes to construct an educational workshop on some topic. In MOBILE, we call the software configuration for the event a session. For example, one may develop sessions for workshops on computer security, CSI software development, or mobile application development.

The instructor first sets up a base installation of the MOBILE environment. Then, the instructor configures the session by installing any necessary application software that either runs directly on top of MOBILE or on top of a virtualization layer included in MOBILE and based on the open-source Oracle Virtual Box product [21]. The instructor may not want to teach in an Ubuntu Linux environment; this virtualization layer allows other operating systems to be used. Virtual machine images and application software for any operating system can be included, subject to licensing rights.

Next, educational materials such as slides, lab exercises, papers, and so on can be loaded. The virtual images, plus any application software, plus any instructional materials, are considered to be content, which is distributed to each participant along with the base MOBILE platform.

Once the session is configured by the instructor, the instructor clones onto other storage devices as many copies as necessary for the number of participants in their instructional setting.

The overall architecture of MOBILE is shown in Figure 1.

### 3.1.2 Networking

During a MOBILE session it will almost always be useful for the instructor computer and all participant computers to be networked together. This allows the addition of educational communication features such as “hand-raising,” chat between participants and the instructor, and screen captures, which MOBILE supports through another Open source web conferencing software package called Big Blue Button [4]. Networking also allows for interaction between participants (e.g., gathering information from remote machines in a computer security cyberwar exercise), as well as between a given participant and the instructor (e.g., remote instructor examination and/or testing of each participant-written mobile application).

A base networking layer is considered to be outside of the scope of MOBILE per se, so if the session demands networking, an existing network must be present and all participating systems must be able to access this network.

In the case of a session which demands isolation (e.g., a cyberwar exercise), an isolated network can be created in a variety of ways, such as by using a switch and network cables, or by using a session-specific wireless access point which can be accessed by each participant computer.

### 3.2 Workflow

#### 3.2.1 Session Preparation

As noted in Section 3.1, the instructor must set up a number of layers for a MOBILE session. These layers are discussed in more detail below, and a summary of the steps is shown in Figure 2.

First, the base MOBILE platform must be created. This is done by downloading the alternate install version of the current Ubuntu Linux release. This version is necessary so that the actual installation can be run with a custom installer configuration file provided by the MOBILE developers, which customizes the installation to provide the MOBILE base and adds some necessary base applications (e.g., the Virtual Box virtualization layer).

Second, content specific to the given session must be created and added to the MOBILE content area. This includes virtual operating system images, which in turn must have any necessary applications installed as well. Such virtual operating system images can be created by the instructor, or obtained from virtualization repositories such as Jumpbox [2].

Third, the instructor needs to either clone enough copies of the session for each participant onto portable read-write media such as a USB disk, or make the MOBILE session...
Pre-Workshop Checklist

1. On base machine, install Ubuntu Linux from alternate installer using MOBILE’s custom installer configuration file.

2. Prepare on base machine workshop content, including virtual machine images, slides, and supplementary software.

3. Clone base machine on to portable storage devices or to a downloadable system image.

4. Distribute clones to attendees.

Figure 2: A summary of the steps an instructor must complete to administer a MOBILE-based workshop.

available for download. Utilities for cloning and updating MOBILE environments are included in the platform.

Fourth, the instructor needs to distribute the cloned MOBILE platform to each participant, which can be done easily just prior to the session itself if cloned drives are used. If a downloadable system image is created, the participants are responsible for downloading MOBILE to a portable storage device before the workshop.

It should be noted that both distribution options have advantages in certain situations. While the former option of making physical clones is more time-consuming for the instructor, it ensures that each participant will have the system ready to go when the session starts. The latter option of making a downloadable image is easier for the instructor, but may be time-consuming for each workshop participant depending on the size of the image and the local network bandwidth, and it is of course essential that each participant complete this step before the session.

3.2.2 Session Process

As a networked MOBILE session begins, the instructor first boots the MOBILE environment and runs a utility called the presenter to establish a session. Each participant then boots their MOBILE environment and runs a utility called the announcer, which allows the participant to specify the name of the desired session and announces their participation in that session. The instructor’s presenter window shows the participants as they connect to the session, which then enables further communication and interaction.

MOBILE supports several additional actions during the session. First, MOBILE supports the copying of files from the instructor to each participant system (e.g., a modified set of slides or an updated lab exercise), though this is limited in practice by the size of the files and the network bandwidth (e.g., copying an updated client OS image to thirty participants would not be feasible in most situations.) Second, MOBILE supports the remote execution of scripted commands on a participant system. This can be used for actions such as the automated installation of a software package on the client systems in order to guarantee a certain configuration, or the remote execution of commands on the participant system to check that system’s status or change the system’s configuration to a known state.

It is also possible to further customize a session environment. For example, in a cyberwar exercise, you may want to have other host systems in the network act as possible targets and track how each participating system meets a certain set of session requirements (e.g., keeping up a specified set of software services). While we have proven the concept by making such customizations in specific sessions, generalizing support for such customization is part of our future work.

At this point, a workshop can proceed, with participants working individually in their MOBILE environment, but communicating with the instructor and other participants as needed as the workshop progresses.

4. RESULTS

We have used a prototype version of MOBILE in computer security workshops for technology instructors at the annual ACM SIGCSE conference in the years 2005, 2006, 2007, and 2009. Each workshop consisted of two stages: the first focused on information gathering and vulnerability assessment, and the second on system hardening and a small cyber-defense exercise which involved a period of system hardening followed by a period of scripted attacks from the workshop organizers. The workshops contained a total of seven exercises.

The first three years’ workshops used team-provided notebook computers and a wired/switched network, while the last year’s workshops used participant-provided notebooks. We provided two virtual machine images—a Windows image and a Linux image, both further configured with various security tools. Participants learned about and practiced using the tools during the workshops, and they then applied their overall knowledge in the final cyber-defense exercise. We further enhanced the environment by adding additional target systems to the network to provide different levels of secured systems for participants to investigate, and we added additional software utility support for tracking the availability.
of services on each participant team as well as for scoring the overall exercise.

Each exercise consisted of two parts. The first was a more traditional lecture based on slides relevant to one computer security topic such as information gathering or system hardening as well as sample Linux and Windows security tools that were useful for that topic. The second was a hands-on exercise, where participants were given a lab sheet with several focused questions, leading them to practice with the relevant tools. The exercises involved a combination of tool usage local to each participant’s system, as well as tool usage across the session network which interacted with the target systems and/or other participants. The MOBILE environment supported these hands-on exercises very well.

We had no system failures during these four workshops, though we discovered issues with individual participant machines due to their lack of sufficient main memory or other resources. Precisely identifying participant system resource needs is part of our continuing work.

Our research team has also created several other session scenarios as examples for future workshop and general use. These scenarios include CSI software development using Java and Eclipse, mobile application development using the Android platform, and teaching graphics exercises using GLUT and OpenGL. A body of images supporting common session scenarios will be made available in a repository to support the overall exercise.

Overall, we have found that the MOBILE system enables hands-on CS workshops in a flexible yet stable environment. Preliminary usage shows that MOBILE can meet the goals and deal with the problems that originally led to this project.

4.1 Assessment

We have not yet directly assessed the effectiveness and usability of the MOBILE system. However, our assessment of the computer security workshops using the MOBILE prototype was very positive and provides initial evidence of its usefulness. A chart of selected evaluation questions and scores is shown in Figure 4. Overall evaluation of the workshops certainly includes evaluation of the content and instruction as well as the MOBILE platform. However, we feel that the positive response to the question asking if the workshop participants want to use our presentation tools does give initial evidence of the quality of our prototype system. An area of future work is indicated by the need to make the system more usable for the average information technology instructor.

We are ready to begin giving dissemination workshops with a full version of the MOBILE system starting in the summer of 2012. We plan for specific assessment of 1) overall effectiveness of MOBILE system in delivering quality hands-on information technology education to instructors, and 2) the usability of the MOBILE system in preparing, distributing and delivering educational materials to workshop participants. Focused assessment in these areas will better evaluate the quality, usability, and overall usefulness of the MOBILE system.

5. FUTURE WORK

While we have accomplished much, there is much more to do as well. Our immediate focus involves the following.

First, we will further develop the MOBILE system functionality and interface, with an emphasis on increasing the usability of the system. Ease of use is key for gaining wider usage of the MOBILE system. Further assessment of usability enhancements is essential and will be done as the interface is finalized. Providing an application programming interface or other means of customizing session functionality is a priority improvement as this allows for a better fit of a given topical session to many possible desired workshop environments and instructor goals.

Second, we plan for further dissemination through faculty workshops starting summer 2012. The dissemination workshops will also be used to find additional use cases which feed back into our development cycle. While we have found many possible uses and several potential development paths for the MOBILE system, we want to get more feedback on the best ways of making the MOBILE system useful to information technology instructors.

Third, we plan for specific assessment as noted above. While the initial evidence of usefulness is promising, we need to verify that the MOBILE system provides useful functionality in a package that is usable by the average information technology instructor. Assessing both functionality and usability are necessary to accomplish this goal.

Fourth, we are working to release, distribute, and support the MOBILE system as an open-source project, which will make the system available to all to use, modify, and enhance. We are also working to develop a community of users that can carry on development of this system after our research work ends.

Fifth, there are potentially other directions for the MOBILE system. Future hands-on information technology education will likely be occurring in the cloud or in some kind of virtual laboratories, and we will examine how the MOBILE system might fit with these new directions as well.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DUE-0817295. We also acknowledge the design and development work performed by entire MOBILE team, including Tom Paine, Jamison Schmidt, Jason Wudi, and numerous student developers.
7. REFERENCES