The UbiLAB Framework for Remote Laboratories

Marija Kalendar[1*], Zivko Kokolanski[1], Branislav Gerazov[1], Gorjan Nadzinski[1], Marija Poposka[1], Aleksandra Zlatkova[1], Filip Donchevski[1], Marija Markovska[1], Anastasiia Sapeha[2], Kirill Karpov[2], Danijela Efnusheva[1], Dusan Gleich[3], Andrej Sarjas[3], Blaz Pongrac[3], and Eduard Siemens[2]

[1] Faculty of Electrical Engineering and Information Technologies, Ss Cyril and Methodius University, Skopje, R. N. Macedonia
[2] Department of Electrical, Mechanical and Industrial Engineering, Anhalt University of Applied Sciences, Köthen, Germany
[3] Faculty of Electrical Engineering and Computer Science, University of Maribor, Maribor, Slovenia
*marijaka@feit.ukim.edu.mk

***Abstract* — Starting from the period of the Covid-19 pandemic, when the presence of students in the university laboratories was very limited, the research work on remote laboratories significantly increased. The most affected aspect of students' study experience at that time were the laboratory experiments intended to be conducted on-premises, in the university laboratories. The UbiLAB framework was envisioned as a digital space for implementing and conducting laboratory work and experiments that are rarely completely transferred into the digital world. Here we have identified two most difficult challenges: achieving a nearly realistic substitute for learning experiences in the actual physical laboratories, and a nearly realistic substitute for the social element of collaborative learning and "making friends" in the process. The innovative cloud-based UbiLAB framework addresses both challenges, digitizing various and diverse actual laboratory experiments through a simple and widely available platform, while enhancing the reality of students' experience in the process.**

Index Terms—Remote laboratories framework, collaborative learning, cloud implementation, LMS systems, remote and virtual laboratories

I. Introduction

Hands-on experience in a laboratory setting is crucial for students pursuing degrees in certain fields of study, especially electrical engineering. It allows students to apply theoretical concepts to real-world scenarios, providing them with practical skills and abilities, and increasing their understanding of complex systems. This helps prepare them for careers as engineers, researchers, and future innovators. On the other hand, this can be hindered by lack or limited access to lab facilities, caused by limited resources (space, equipment, staff), as well as time or financial constraints.

The lack of physical access to educational facilities was exacerbated during the COVID-19 pandemic. The ensuing lockdowns caused a paradigm shift of education from on-site to online. While theoretical lessons can be effectively delivered using online teaching platforms, the change poses significant challenges when it comes to accessing laboratory equipment and resources remotely.

Remote Access Laboratories (RALs) were initially established in Australia and Switzerland in 1994. Traditionally, Australian RALs have focused on electrical, electronic, and computer engineering, while European RALs have also included mechanical engineering and mechatronics [1]. Gustavsson proposed a RAL for electrical circuit experiments [2]. RAL frameworks for control engineering implementing SCADA and programmable logic controllers (PLCs) have also been proposed [3, 4]. The social interactions that are also an important part of the learning process have been researched [5], integrated into RAL technologies with collaboration-based eLearning [6] and with focus on VET application [7]. While RALs usually follow a centralized client-server paradigm, there are also peer-to-peer frameworks which allow hosting experiments at distributed locations, allowing the students to be both users and creators [8, 9]. One of the most successful RAL frameworks is Weblab-Deusto [10]. The framework's software is available under an Open Source license and the hardware equipment can be duplicated. This has led to the framework being used in 10 educational institutions in 7 countries across the world.

In this paper we outline the results of the Erasmus+ project "A ubiquitous virtual laboratory framework – UbiLAB". It proposes an innovative framework designed to support endpoints such as hardware devices, virtual devices, or software solutions. It presents a foundation for a rich and diverse set of laboratory experiments with modules supporting software and hardware exercises. It is also licensed as free software, for increased impact in the education community.

The project activities researched the previous approaches for RAL designs [6], [9]-[14], identified the strengths and weaknesses [15], and finally proposed an architecture design for the UbiLAB framework.

The design of the UbiLAB framework architecture depicted in Fig. 1 presents the three main elements: hardware/physical based laboratories, software/virtual based laboratories, and the core cloud modules (Core Cloud backend). The latter enable different digitalization possibilities, common student activities tracking, scheduling of laboratories, collaboration modules.

As a result from the research conducted in [15] and in the direction of open science as an EU Commission goal, the UbiLAB framework was based and built on top of the widely used open source Learning Management System (LMS)

This work has been supported by the European Erasmus+ project "A ubiquitous virtual laboratory framework" (UbiLAB, ERASMUS+ Key Action 2, ref: 2020-1-MK01-KA226-HE-094548).

Moodle and Open edX, using and modifying some of the preexisting modules in the context of the UbiLAB framework.

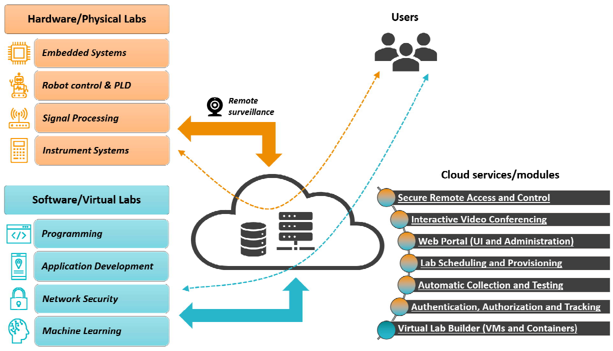

Fig. 1. The Ubilab Framework Architecture.

II. Cloud Services and Modules in the UbiLAB Framework

The core cloud backend of the UbiLAB framework comprises the services and modules that enable remote access to laboratory resources, virtual presence during the lab exercises, as well as online interaction with colleagues and instructors, thus enabling an environment that provides a virtual experience as close as possible to the physical presence in the actual laboratories. The core modules and the main provided functionalities are the following:

- **Web portal** for connecting, accessing and control of all framework resources.
- **Secure Remote Access and Control module** for remote and secure access to the framework resources.
- **Authentication, Authorization and Tracking module** for user authentication and authorization and interoperability between different sign-on systems, while tracking and controlling resource access.
- **Lab Scheduling and Provisioning module** for class, time slot and experiment scheduling, laboratories monitoring and provisioning, and resource utilization.
- **Interactive Video Conferencing module** for direct communication between students and teachers integrating different video conferencing tools.
- **Automatic Collection, Testing and Checking module** for student results, projects and online exams.
- **Virtual Lab Builder module** for building complete virtual laboratories by using virtual machine, container management and orchestration tools to deploy virtual software environments.
- **Video Monitoring module** that uses web cameras from physical labs and feeds video streams to students, teachers and administrators.
- **Collaborative learning experience module** implementing tools in any laboratory that facilitates the group learning experience: sharing results, comments, communication, jointly reaching solutions and exchanging knowledge.

III. Virtual Software Laboratories in the UbiLAB Framework

Software-based laboratories as part of the UbiLAB framework enable development of online modular software-based laboratory exercises. The Virtual Software Laboratories (VSLs) can be implemented either on physical computers located at university premises or virtual environments running on dedicated servers, provided by the core cloud backend (Virtual Lab Builder module) of the UbiLAB framework. Additionally, the core cloud backend is providing support for modular configuration of these VSLs by utilizing configurable software modules, virtual machine images, and containers, building the needed specific environment for the experiment.

The VSLs comprise the software-based remote laboratory exercises and provide the students with access to readily prepared environments, with no technical requirements on the client side. Consequently, the VSLs are used for designing various software-based laboratory exercises, as presented later. Additionally, the designed VSL based laboratory exercises accommodate a new organizational concept by enabling more tools for local self-assessment for the students, with the ability to evaluate their work and identify underlying problems before submissions.

A. Virtual Software Laboratories Setup

The VSL developed via the UbiLAB framework provide a seamless and accessible online learning experience for users by designing a sophisticated setup integrally with the most widely used LMSs in university education, Moodle and Open edX. The general setup of the core cloud backend, that allows the students to access and practice the exercises remotely while ensuring a secure and stable environment, is presented in Fig. 2. The elements of the designed UbiLAB framework core cloud backend comprise:

Proxy Server, which acts as a barrier between the Internet and the internal network, providing essential security and accessibility, and hiding the designed high-performance infrastructure from the open world.

Guacamole Services and LMS Systems, which are accessible through the proxy server. The LMS serves as a stable platform for delivering online learning content, while Apache Guacamole is an open-source remote desktop gateway with a web browser interface that facilitates connections to the high-performance server or the remote computers in the physical laboratories where the remote software laboratory exercises will take place. This eliminates the need for complex software installations or network configurations, making it easy for users to engage directly with the already set-up VLS.

Remote Virtual Software Environment (VSE), which could be any of the available software environments connected in the network (different operating systems on connected university computers, virtual machines or software containers hosted and managed by the core cloud backend). The core cloud backend acts as a high-performance server, running docker containers or virtual machines, depending on the needed software set-up for the exercise and the available resources. The container management environment enables creating temporary instances for each student, thus offering a highly scalable and flexible infrastructure to accommodate a large number of online learners.

iFrame (inline frame) Integration, for integrating the access to the Guacamole Server into other web pages. Embedding the Guacamole web interface within an iFrame in the LMS of choice allows access to the remote VSEs or applications directly within the LMS platform, providing a unified and comprehensive learning experience through the UbiLAB framework.

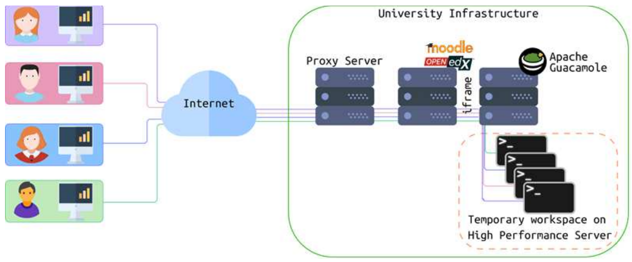

Fig. 2. Virtual Software Laboratories setup.

B. Virtual Software Laboratories examples

One example of an implemented VSL is the **Operating Systems Lab**, designed to be completely self-paced, step-by-step guided, and with practical exercises implemented on specially configured Docker containers for VSEs. Despite the self-learning environment, and the VSEs that enable virtual environments for doing the practical given tasks, this laboratory includes automatic exercise checking scripts in order to evaluate the students' progress and keep information about it in the UbiLAB framework.

The **Remote Virtual Instrumentation Laboratory (RVIL) platform** is another example of an implemented complete VSL that enables engineers to conduct a variety of experiments using virtual instrumentation (multimeters, power supplies, oscilloscopes, function generators, data acquisition cards), overcoming the need for specialized and expensive laboratory equipment while providing students with an almost realistic user experience. The virtual instruments use network-shared variables that can be connected to any kind of device (virtual or real), database or external resource through the Internet. It is integrated into the UbiLAB framework using the VSE approach by remotely accessing computers and virtual machines hosting this RVIL through Apache Guacamole.

Hardware support is also available for integrating any National Instruments-related hardware or Arduino boards into the virtual experiments. The platform also supports specific digital protocols, such as I2C, SPI, UART, and PWM.

There are also benefits that RVIL offers over traditional physical laboratory setups, by allowing for easy sharing and replication of experiments and collaboration, as users can access the same virtual instruments and setups regardless of their location. It can also be used as a teaching tool to enhance engineering education. By providing a virtual instrumentation laboratory environment, the platform offers an interactive and engaging learning experience without the risk of damaging physical equipment or consuming expensive resources.

IV. Hardware and Augmented Reality Laboratories in the UbiLAB Framework

The hardware-based remote laboratories are focused on facilitating the realization of exercises from courses offered in the areas of Embedded systems, Programmable Logic Controllers (PLCs), Microcontrollers, Field-programmable Gate Arrays (FPGA), Electrical Measurements, and Digital Signal Processing (DSP). These experiments are already set-up in the existing and well-equipped laboratories, but virtualization of the laboratories was facilitated by integrating into the UbiLAB framework: a video stream adaptation for monitoring the used laboratory hardware; a remote access module to interface with the PCs connected to the hardware on-site; the aforementioned RVIL platform; and an augmented reality (AR) module enabling virtual object insertion in the video monitoring stream from the lab.

These additions enabled the design and transfer into the UbiLAB framework of the following existing hardware-based laboratory exercises.

A. DSP hardware laboratory remote experiments

The DSP hardware set-up encompasses multiple NI myRIO 1900 devices, hosting a different experiment each, applied for real-time audio analysis and signal processing (spectral analysis, time-frequency analysis, filtering and signal generation).

These experiments utilize LabVIEW for the user front-end interface and remote control, so they were integrated by utilizing the remote access module to interface directly with the PCs connected to the hardware on-site. This gives the remote user direct access to the UI and the myRIO software and hardware, and enables them to change predefined parameters and observe the response in numerical, graphical and audio format, allowing for the same possibilities as if the experiments were performed physically in the laboratory. Additionally, a Zoom connection is integrated to enable the students to process the audio signal streamed from their microphone. Hence, the student remote side requires only a microphone (for real-time audio streaming) and headphones (for hearing of the result by the processing of sounds).

B. Embedded hardware laboratory remote experiments

The Embedded hardware laboratory provides students with practical introduction to the field of embedded systems, allowing them to practice designing and controlling embedded hardware. The setup enables access to six laboratory exercises designed to be done in a single session encompassing: general purpose I/O ports, general purpose timers, analog/digital convertors (ADC), universal asynchronous/synchronous receiver/transmitter (USART), and inter-integrated circuits (I2C) communication. Gradually finishing all exercises and incorporating the respective hardware results in the design of a working autonomous irrigation system.

The connected hardware setup (given in Fig. 3) comprises a PC, STM32 microcontroller, LED diodes on different digital pins of the microcontroller, SEN0114 soil moisture sensor connected to an analog pin, SHT20 - temperature and humidity sensor connected to I2C compatible pins, a USB to TTL adapter connected to the computer for USART communication, and a camera for observation of the experiment. This setup is again integrated within the UbiLAB framework by utilizing the remote access module to interface directly with the PCs connected to the hardware on-site. The remote users can program and control the hardware, and observe the results through the serial monitor (text-based) or through the live video stream.

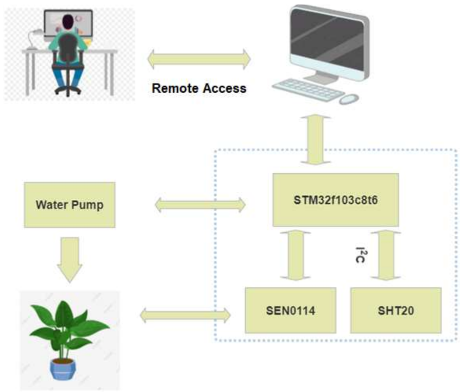

Fig. 3. Remote embedded laboratory setup.

C. *Network Control Systems laboratory remote experiments*

The Network Control System (NCS) laboratory remote experiments integrated into the UbiLAB framework present two different use-cases: a remotely controlled wind levitation system, and a dual temperature control system. Both are based on an STM32F767 microcontroller, both enable students to model the system and design and test two a PID and a Super Twisted controller. Fig. 4 shows the GUI developed for the wind levitation system.

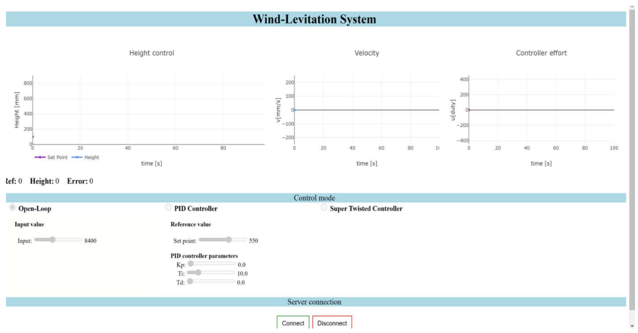

Fig. 4. Developed GUI and WebSocket client using HTML and Javascript.

The remote access is facilitated by software components integrated in the UbiLAB web interface connected directly to the hardware via network tunneling. The local computer (gateway) serves as a connection between the embedded system and the designed WebSocket Python server. The WebSocket client can directly connect to this server to gather data and provide new control parameters. The server-client communication is secured using an SSL certificate and safety measures such as communication loss detection and emergency shut-down. Once the client receives the data from the embedded system, the data representation to the user can be designed in Javascript, MatLab, Simulink, LabVIEW, or Python, and then presented on a suitable graphical user interface, as shown in Fig. 5.

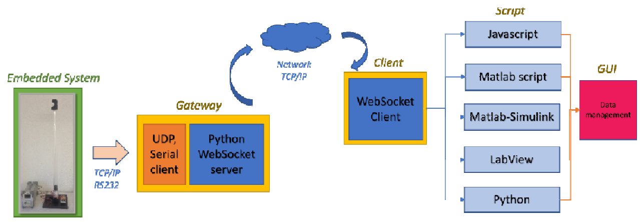

Fig. 5. NCS control system for remote laboratories.

D. *PLC laboratory and Augmented Reality remote experiments*

The Programmable Logic Controllers (PLC) hardware setup simulates an industrial application and a developed augmented reality module essential for virtualization of lab exercises that include manipulation of physical objects. This hardware laboratory setup encompasses a computer, PLC hardware, suitable sensors, and a conveyor belt. The remote access module again interfaces directly to the on-site PC. This PC is used to program the PLC to control the conveyor belt connected to it. The conveyor belt includes four digital proximity sensors (three inductive and one photoelectric), four push buttons, a DC motor for the conveyor, and an analog potentiometer. Thus, the setup requires a total of 8 digital inputs, 1 digital output, and 1 analog input from the PLC. Students are required to develop a program for the PLC in order to control the conveyor belt and afterwards test and observe its operation.

The on-site experiment involves students' interaction with the conveyor belt by placing metallic and non-metallic objects on the conveyor and working with the push buttons or the analog potentiometer, and direct observation of the results. In the remote setup of the experiment, the augmented reality module implemented in the UbiLAB framework web interface enables interaction with virtual objects (boxes, bush buttons, potentiometer), presented in the UI and overlapped with the remote live video stream showing the conveyor. The web application utilizes the AR module and enables the virtualized interaction together with a specific communication protocol in order to send the outcome of the interaction to the PLC. The PLC program will enable the physical system to react accordingly to the interaction in the same way as physical objects were placed or buttons were pushed. The students can observe the results through the live video stream.

V. Conclusion

With the pandemic coming to an end, the intent to digitalize the laboratories still proves to be a significant goal and digital remote laboratories can also find application in enabling digital remote experiments to high school students or enabling remote access to laboratory equipment and experiments to persons from remote and hard to access environments. The results presented here show that UbiLAB has successfully designed a framework for remote laboratories which will improve students' experience and opportunities in the modern post-pandemic reality of digital education.

References

[1] H. Ku, T. Ahfock and T. Yusaf, "Remote access laboratories in Australia and Europe," European Journal of Engineering Education, 36(3), pp.253-268, 2011.

[2] I. Gustavsson, "A remote access laboratory for electrical circuit experiments," International Journal of Engineering Education, 19(3), pp.409-419, 2003.

[3] Z. Aydogmus and O. Aydogmus, "A web-based remote access laboratory using SCADA," IEEE Transactions on education, 52(1), pp.126-132, 2008.

[4] J. S´aenz et al., "Open and low-cost virtual and remote labs on control engineering," IEEE Access, vol. 3, pp. 805–814, 2015.

[5] L. de la Torre Cubillo et al., "Providing collaborative support to virtual and remote laboratories," IEEE Transactions on Learning Technologies, 10 2013.

[6] M. Helander and M. Emami, "Engineering eLaboratories: Integration of remote access and eCollaboration," International Journal of Engineering Education, 24(3), p.466, 2008.

[7]]D. Gleich, B. Gergic, S. Temkova, D. Georgiev, Z. Kokolanski, T. Suminoski, S. Simovic, M. Podobnik, Z. Ruscic, M. Pavosevic "CORELA: Collaborative Learning Platform With Integrated Remote Laboratory Environment in VET", International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 151-154, Osijek, Croatia, 2019

[8] A.A. Kist, A. Maiti, A.D. Maxwell, L. Orwin, W. Midgley, K. Noble, and W. Ting, "Overlay network architectures for peer-to-peer Remote Access Laboratories," In 2014 11th International Conference on Remote Engineering and Virtual Instrumentation (REV), pp. 274-280. IEEE, 2014.

[9] A. Maiti, A.A. Kist, and A.D. Maxwell, "Real-time remote access laboratory with distributed and modular design," IEEE Transactions on Industrial Electronics, 62(6), pp.3607-3618, 2015.

[10] P. Orduña, J. Garcia-Zubia, L. Rodriguez-Gil, I. Angulo, U. Hernandez-Jayo, O. Dziabenko and D. López-de-Ipiña, "The weblab-deusto remote laboratory management system architecture: achieving scalability, interoperability, and federation of remote experimentation," Cyber-Physical Laboratories in Engineering and Science Education, pp.17-42, 2018.

[11] I. Gustavsson et al., "On objectives of instructional laboratories, individual assessment, and use of collaborative remote laboratories," IEEE Transactions on Learning Technologies, vol. 2, no. 4, pp. 263–274, 2009.

[12] D. Galan et al., "Automated assessment of computer programming practices: The 8-years uned experience," IEEE Access, vol. 7, pp. 130 113–130 119, 2019.

[13] Y. Chaabi et al., "Development of a learning analytics extension in open edx," in 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI), 2021, pp. 1–6.

[14] N. Simanullang et al., "Learning management system (lms) based on moodle to improve students learning activity," Journal of Physics: Conference Series, vol. 1462, p. 012067, 02 2020.

[15] A. Sapeha, et. al, "Learning management systems as a platform for deployment of remote and virtual laboratory environments," Proceedings of International Conference on Applied Innovation in IT, vol. 10, 1, pp. 133-142, 2022.

[16] Ž. Kokolanski, B. Velkovski, T. Shuminoski, V. Dimcev, D. Taškovski "Collaborative Learning Platform with Integrated Remote Laboratory Environment in Vocational Education", Journal of Electrical Engineering and Information Technologies, pp. 117–123, 2021