

Design of a System to Track Student Progress in Virtual Laboratories

Tilman Wolf and Rui Yang
Department of Electrical and Computer Engineering
University of Massachusetts Amherst
{wolf,ryang}@ecs.umass.edu

Abstract—Laboratories are an important part of technical and scientific education. To amortize the cost of setup and maintenance, some laboratory facilities can be shared and via remote network access. Such virtual laboratories provide realistic environments and allow for interactive student experimentation. When students access a virtual laboratory via the Internet, it makes it more difficult for instructors to track student progress. In this work, we present the design of a system that can track student progress as an extension of the Open Network Laboratory. Our design presents a mechanism for recording student work, comparing it to a master solution, and visualizing the status of the entire class. This system helps simplify the teaching of courses using the Open Network Laboratory and virtual laboratories in general.

I. INTRODUCTION

Advanced technical education requires a solid foundation of theoretical knowledge, but also training in state-of-the-art techniques, tools, and applications. Such skills are often taught in laboratory settings where students can experience technology hands-on. Since setup and maintenance of laboratories can be expensive, there has been a trend towards “virtual laboratories.” In these virtual laboratories, students can experiment with real systems and technology (i.e., not just simulation) via remote access. A single virtual laboratory can serve a large community of geographically distributed students and amortize setup and maintenance cost.

Our work focuses on one such virtual laboratory, the Open Network Lab (ONL), which has been set up by our collaborators at Washington University in St. Louis [6]. ONL hosts a number of real, interconnected network routers and end-systems. Through a graphical user interface, students can configure the system for their particular experiments, log into computers, send real network traffic, and record measurements. In the context of this system, a number of educational challenges arise. In particular, we look at the problem of how to track student progress. In conventional laboratories, the instructor can “look over the shoulder” of students to identify who needs help. In an online setting, this is much more difficult since students access the laboratory remotely from different locations and at different times.

In this paper, we present the design of a system that allows instructors to track student progress in ONL. The main idea is to use a “session recording” feature that permits students to save their progress to a central server. Using this information, “session analysis” software compares each student’s work automatically to an exemplary laboratory session provided by the instructor. The differences between the students’ progress and a master solution is stored in a database and made available to the instructor on demand. Using a visualization of student progress, an instructor can easily scan the progress of the entire class and identify students who need further help. The specific contributions of this paper are

- the design of a progress tracking system and its integration into the ONL system architecture,
- a process for comparing student progress to a master solution,
- a visualization of student progress that shows the status of large numbers of students in a single display, and
- a prototype implementation of the tracking system specific to ONL.

While our design is described in the context of ONL, it is applicable to other virtual laboratories, too.

The remainder of the paper is organized as follows. Related work is discussed in Section II. We briefly describe the Open Network Laboratory in Section III. The design of the session recording and analysis system that we use to track student progress is presented in Section IV. Our prototype system is briefly described in Section V. Section VI summarizes and concludes this paper.

II. RELATED WORK

It has long been recognized that laboratory experiences are important aspects of engineering education [5]. Laboratory education aims at providing students with experience related to the theory and practice of experimentation [7]. In the sciences and in computer science, these experimental skills are considered an essential preparation for the workplace [15]. One important aspect of laboratory experimentation is to expose students to

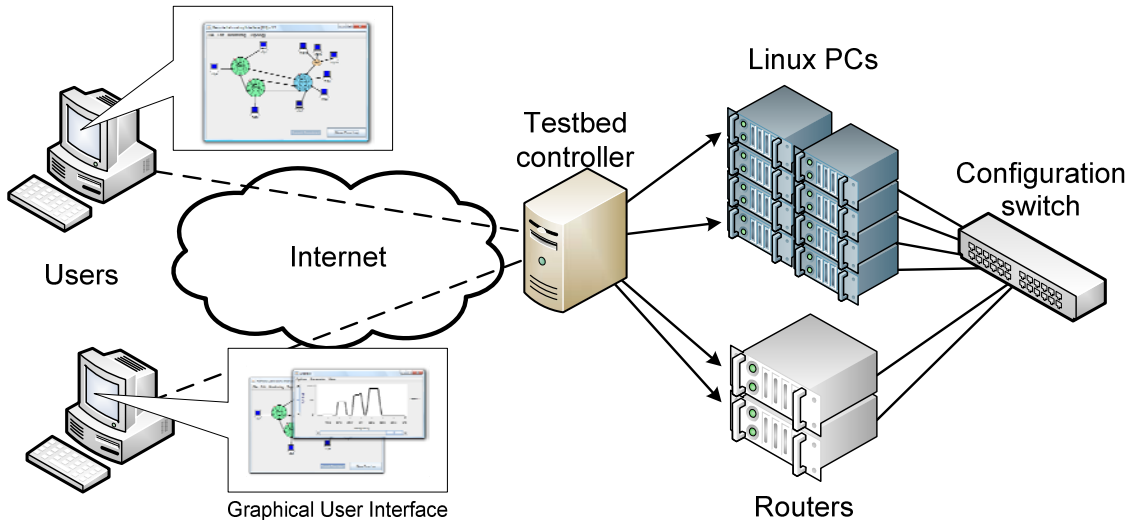


Fig. 1. ONL System Architecture.

realistic settings, where problems and failures can occur (e.g., in network security [11]). While simulators may provide a realistic perspective on a system, they often do not model failures in a sufficiently realistic way [9].

Virtual laboratories provide remote access to real experimental facilities and thus provide access to a large community of users. One such virtual laboratory is the Open Network Laboratory (ONL) [6] used in this study. User can access the facility, set up and control experiments, and collect measurement data via an easy-to-use graphical user interface. Virtual laboratories have been developed for other technical areas: thermodynamics [8], civil engineering [22], geotechnical engineering [21], and cell biology [16]. Related technical and educational issues have been explored in the context of access to educational resources for distance education [1], [2].

The Open Network Laboratory used for this study has been design for education in computer networking. There exist other virtual and remotely accessible lab facilities for computer networks (e.g., Emulab [17] and Planetlab [14]), but they are mainly designed for academic research and are more complex to use. In prior work, we have reported on our teaching experience with ONL [20] and operational issues related to using and maintaining such a facility [18]. In our recent assessment study, we observe that the number of learning events that occur while using ONL is nearly as high as in traditional lecture settings [19]. Other assessment studies have shown the effectiveness of virtual laboratories for civil engineering [22], geotechnical science [21], and computer networks [12].

The automated analysis of student work that we propose in this paper has been studied previously in the context of grading. Virtual laboratories with automated

grading features have been proposed for computer architecture [3] and electronics [13]. Other automated grading systems have been developed in many other fields and are widely deployed [10]. Our work focuses on how to design a comparison (i.e., grading) system specifically for ONL and how to use the obtained information to allow instructors to help students while they are working on the project before the final work is turned in.

III. OPEN NETWORK LABORATORY

Before discussing in Section IV how we track student progress, we present a brief overview of ONL.

A. ONL System

The Open Network Laboratory is designed to let users experiment with state-of-the-art computer network equipment. The types of systems used in ONL are the same as in the Internet or any data communication network: end-system hosts that run distributed applications that exchange data, routers that forward data through the network, and links that interconnect routers and hosts. When users set up experiments in ONL, they can specify the topology of the network that they want to use (i.e., determine which system is connected to which other system) and they can set up networking-specific parameters (e.g., link bandwidth and delay). Once the topology is in place, the user can log into the end-system hosts and run any desired application. To provide full access, the user has root privileges in the Linux operating system run on the end-hosts. While running applications and generating traffic in the network, the user can continue changing the configuration of the network and collect measurement data on a variety of metrics. What is unique to ONL compared to other testbeds (e.g., Emulab), is the level of system details (e.g., queuing behavior, routing tables,

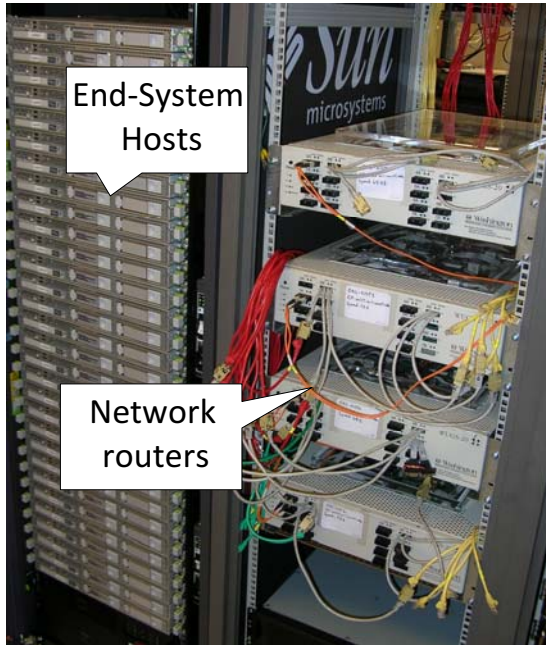


Fig. 2. ONL Hardware.

etc.) that are accessible to the user. This level of realism is essential for a rewarding laboratory experience.

The system architecture of ONL is shown in Figure 1. Users can connect from anywhere on the Internet to the testbed controller. The testbed controller interacts with the user's computer on one side to obtain the network configuration and to report measurement results. Towards the ONL side, the testbed controller connects to the routers and end-system hosts. It also configures the configuration switch to create the virtual topology between routers and hosts that has been requested by the user. Any commands sent by the user are relayed by the testbed controller to the appropriate system in ONL. Figure 2 shows a portion of the physical hardware of the ONL system. On the left, the end-system hosts are shown; on the right, the four router systems are shown.

B. ONL User Interface

ONL provides a graphical user interface for users to interact with the system (see Figure 3). Using this interface, topologies can be created by simply drag and drop of networking components. Also, configuration parameters can be changed by right-clicking on a system component and editing values in a pop-up window. These types of interactions are significantly simpler than the command-line interfaces typically encountered on router systems. Thus, much of the complexity of using network components is hidden to the user, while still providing access to the aspects of the network that are important for the educational experience.

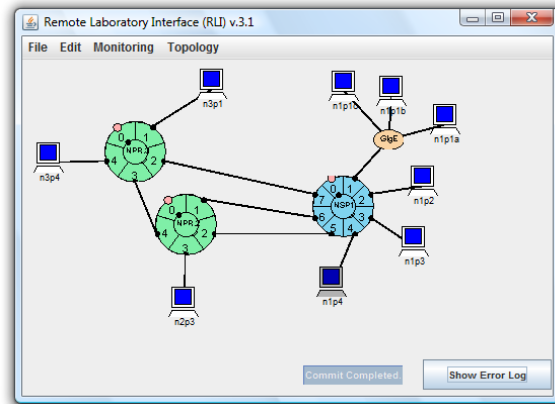


Fig. 3. ONL Graphical User Interface.

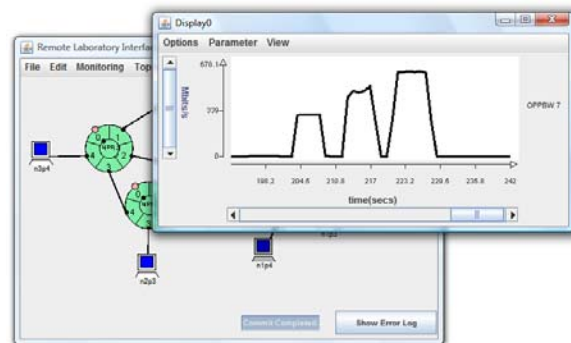


Fig. 4. ONL Measurement Graphical User Interface.

Once an experiment has set up, the user can log in to any ONL end-system host and run any program to generate traffic on ONL the network. To observe what happens in the network, ONL provides measurement windows that graph system metrics on demand. Users can select what they want to observe (e.g., queue length, link utilization, etc.) on any of the ONL systems. An example of a measurement window is shown in Figure 4. The data obtained from ONL can also be downloaded for offline post-processing.

The ONL configuration created on a local machine can be saved to file and restored later. We use this save feature as a part of the session recoding system discussed in Section IV.

C. Teaching with ONL

ONL can be used in a number of different educational settings. The laboratory can be use for laboratory assignments to augment a traditional lecture-based classroom setting. It is also possible to use ONL for distance education. More advanced students also may use ONL for independent research projects. We have discussed some of the teaching experiences with ONL in [20]. The main objectives for using ONL are

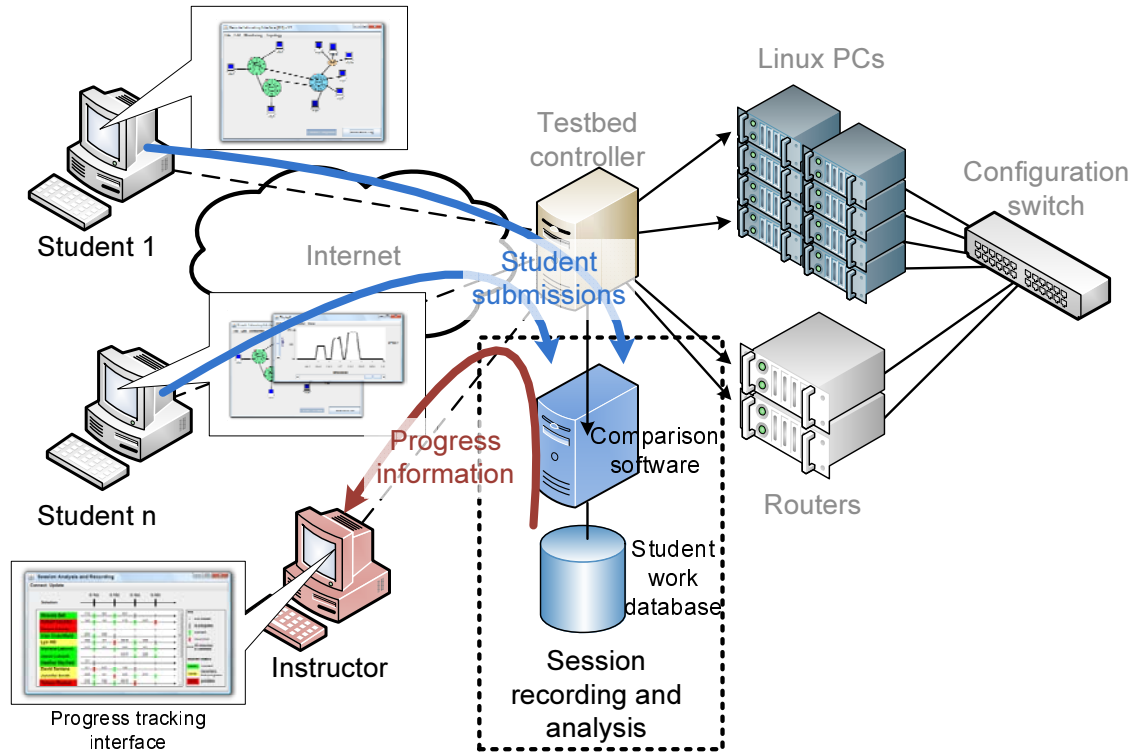


Fig. 5. ONL Session Recording and Analysis Extension.

- to deepen the understanding of networking concepts,
- to develop experimental computer science skills, and
- to increase experience with advanced networking technology.

Due to the easy-to-use GUI, ONL can be used very easily without a deep understanding of network systems. But if desired, ONL also allows for experimentation with very specific details of high-performance routers. Thus, the use of ONL can be adapted to the level of cognitive capabilities (i.e., Blooms taxonomy [4]) that is most suitable for students, ranging from application to analysis, synthesis, and evaluation.

IV. SESSION RECORDING AND ANALYSIS

ONL has been used successfully in several undergraduate and graduate courses at three different institutions. One of the key challenges, however, is to keep track of student performance. This problem applies not only to ONL, but to practically all virtual laboratories. Since the instructor is not able to “look over the shoulder” of students, they cannot easily determine which students are struggling and need help.

In our work, we address this problem and present an extension to ONL to record students’ lab sessions, to automatically evaluate these sessions and compare them

to the master solution, and to report this information to the instructor in an easily understandable representation. The general ideas of this session recording and analysis system can be used for other virtual laboratories if the comparison component is adapted to the specific laboratory environment.

A. Recording and Analysis Architecture

The general architecture of our recording and analysis system is shown in Figure 5. The main addition to ONL is a database that maintains records of student work. Co-located with this database is a computer that performs the comparison of student work with the master solution. The status of all students is reported to the instructor’s computer and displayed.

The detailed recording and analysis process is as follows:

- **Master Solution:** The instructor provides a master solution for an assignment. This solution contains the correct ONL configurations (i.e., saved ONL configuration file) for each question in the assignment. The master solution is stored in the database for comparison with student work.
- **Session Recording:** An extension to the ONL GUI allows students to record “milestones” while solving an assignment. A milestone can be recorded by the student when a question in the assignment

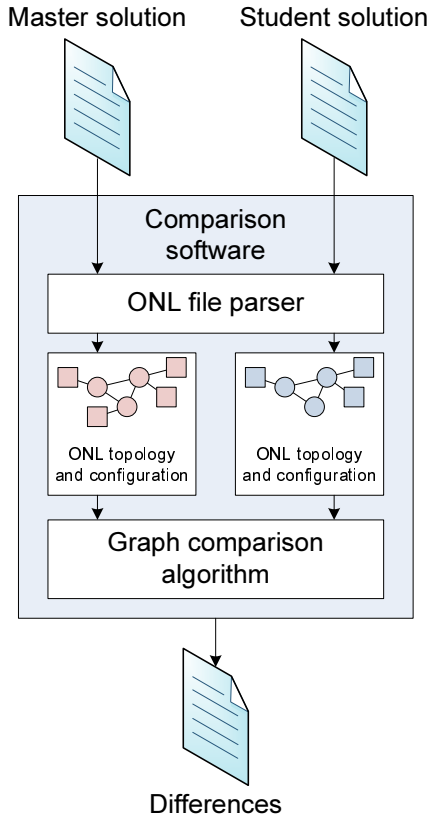


Fig. 6. Comparison Process.

has been completed. The recoding is initiated by the student and can be overwritten if necessary. All (or at least the most recent) milestones for each assignment question are stored in the student work database.

- **Session Analysis:** Whenever a milestone is stored in the student work database, it is compared to the master solution using the automated comparison system explained below. Differences are recorded for reporting to the instructor.
- **Progress Tracking:** An instructor can access the current progress by all students from the student work database. The database reports the results of the most recent session analysis. Using the progress tracking interface discussed below, the overall progress can be quickly examined by the instructor.

For each assignment that is used on ONL, the database keeps separate record. Using appropriate identity management, it is ensured that students and instructors can (only) access the information that is relevant to them.

B. Automated Comparison

The automated comparison software is the main technical aspect of the progress tracking system. The process

TABLE I
ONL PROPERTIES CONSIDERED BY COMPARISON SOFTWARE.

Type of information	Property
Topology	Number of clusters
	Number of end-systems
	Number of links
	Connected ports
Configuration	Routing tables
	Port bandwidths
	Packet filters
	Router queues

is illustrated in Figure 6. The master solution and the student solution under consideration are processed. Note that each milestone in a session is considered a “solution” for the corresponding part of the assignment. Using a custom parser program, each solution file is analyzed and the ONL topology and configuration is extracted. Using a graph comparison algorithm, both topologies and configurations are compared. The differences are reported by the comparison system and stored in the student work database. If there are no differences, then the student solution is considered correct.

The ONL-specific properties that are compared by the system are listed in Table I. The topology comparison is broken down into several aspects to avoid the problem of a general graph comparison, which is known to be intractable. Also, properties that do not contribute to the correctness or incorrectness of a solution are ignored (e.g., system names or placement on GUI). Using this comparison approach, the system can determine if a student indeed has a correct solution.

C. Student Progress Tracking GUI

One of the main aspects of this project is to design a suitable interface through which an instructor can access the information that is tracked by the system. In particular, for courses with large numbers of students it should be possible to identify those students who have problems with a quick glance.

The mockup design of our graphical user interface is shown in Figure 7. The window contains a status report for each student. Each student’s status consists of a logical timeline that contains marks for each milestone of the assignment. In Figure 7, there are two questions with two subquestions each. The key on the right shows the interpretation of each color mark that depicts the status for a particular milestone: green indicates the correct completion of the milestone, red indicates a milestone that does not match the master solution, and gray indicates that the student is currently in the process of solving the problem. The color underlying the student’s name indicates the overall status (across all milestones): green indicates correct solutions up to the point where the student is currently working, red indicates that the

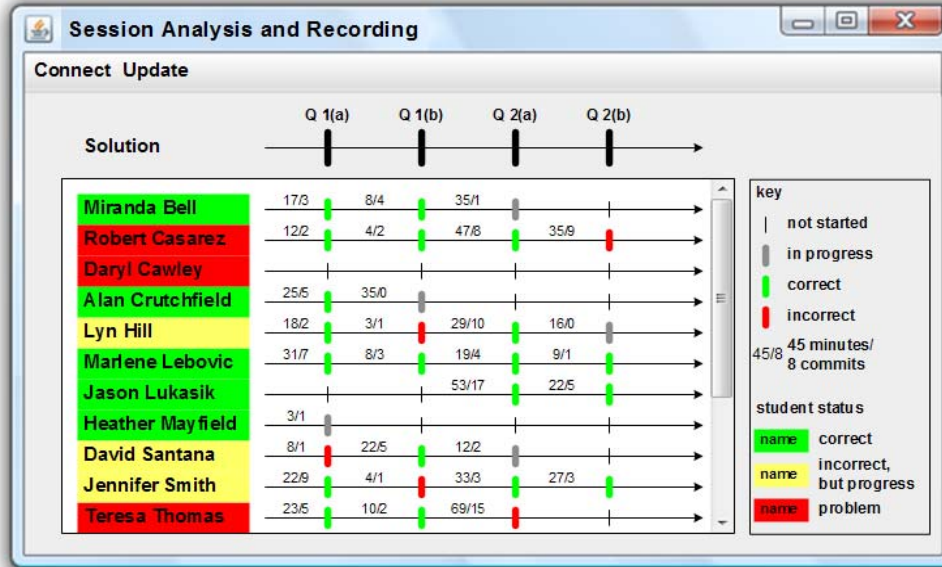


Fig. 7. Mockup Graphical User Interface for Progress Tracking.

most recent milestone is incorrect and thus may be having problems, and yellow indicates that one or more milestones are incorrect but the most recent is correct. In addition, the total time spent and the number of “commits” (i.e., ONL setup commands) is shown for each milestone.

A possible extension for this interface is to not only show if the solution is correct or not, but to display the list of differences (e.g., on mouse-over). It is possible that a difference does not contribute to a wrong solution (e.g., topology contains extraneous host). Thus, the instructor’s insight may be necessary, too.

D. Teaching with Progress Tracking

To use the progress tracking system in a course, an instructor simply needs to call up the GUI to access the student work database. An important property of our visualization design is that it presents the status of all students in one simple view. Thus, it is possible to identify students who have problems (names in red) or who have potential problems (names in yellow). Based on this information, instructors can actively help these students. Our collaborators at Washington University in St. Louis are currently in the process of developing a “session sharing” system, where an instructor can join a student’s session with ONL.

V. PROTOTYPE SYSTEM

We have implemented a proof-of-concept prototype of the system described in Section IV. We use a feature in the ONL GUI that allows the saving of a configuration to record a student’s session. The configuration file is

uploaded to the student work database via a web server that runs a common gateway interface (CGI) script. The comparison software uses a yacc grammar that describes the file format to parse the student milestone as well as the master solution. The topology and configuration comparison algorithm checks the properties shown in Table I. The results of the comparison are currently presented as a text output, but we are in the process of implementing the GUI as shown in the mockup in Figure 7. Also not yet implemented is an identity management system, which is important for a practical deployment. However, the current prototype shows the main interactions of the system and provides a convincing argument that the design is feasible and useful for tracking student progress.

VI. SUMMARY AND CONCLUSIONS

Virtual laboratories that allow student experimentation via the remote access pose a challenge for tracking student progress by instructors. We present the design of an extension for the Open Network Laboratory that records students’ milestones in an assignment, compares their solution to a master solution, and visualizes the progress to the instructor. We believe this design is useful for virtual laboratories beyond the Open Network Laboratory and can help instructor to be more effective.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DRL-0632586. We would like to thank Ken Wong from Washington University for his help in using ONL.

REFERENCES

- [1] B. Aktan, C. A. Bohus, L. A. Crowl, and M. H. Shor. Distance learning applied to control engineering laboratories. *IEEE Transactions on Education*, 39(3):320–326, Aug. 1996.
- [2] A. Bagnasco, P. Buschiazzo, D. Ponta, and M. Scapolla. A learning resources centre for simulation and remote experimentation in electronics. In *PETRA '08: Proceedings of the 1st ACM international conference on Pervasive Technologies Related to Assistive Environments*, pages 1–7, Athens, Greece, July 2008.
- [3] H. Bähring, J. Keller, and W. Schiffmann. A combined virtual and remotely accessible microprocessor laboratory. In *Proc. of the 2004 Workshop on Computer Architecture Education (WCAE) in conjunction with the 31st International Symposium on Computer Architecture (ISCA)*, Munich, Germany, June 2004.
- [4] B. S. Bloom. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longman Group, United Kingdom, 1969.
- [5] Commission on Engineering Education. New directions in laboratory instruction for engineering students. *Journal of Engineering Education*, 58:191–195, Nov. 1967.
- [6] J. DeHart, F. Kuhns, J. Parwatikar, J. Turner, C. Wiseman, and K. Wong. The open network laboratory: a resource for networking research and education. *ACM SIGCOMM Computer Communication Review*, 35(5):75–78, Oct. 2005.
- [7] E. W. Ernst. A new role for the undergraduate engineering laboratory. *IEEE Transactions on Education*, 26(2):49–51, May 1983.
- [8] K. D. Forbus, S. E. Kuehne, P. B. Whalley, J. O. Everett, L. Ureel, M. Brokowski, and J. Baher. CyclePad: an articulate virtual laboratory for engineering thermodynamics. *Artificial Intelligence*, 114(1-2):297–347, Oct. 1999.
- [9] A. Francis and M. Couture. Realism in the design process and credibility of a simulation-based virtual laboratory. *Journal of Computer Assisted Learning*, 20(1):40–49, Feb. 2004.
- [10] D. Hart, B. Woolf, R. Day, B. Botch, and W. Vining. OWL: An integrated web-based learning environment. In *Proc. of International Conference on Mathematics / Science Education and Technology*, pages 106–112, San Antonio, TX, Mar. 1999.
- [11] J. M. D. Hill, C. A. Carver, Jr., J. W. Humphries, and U. W. Pooch. Using an isolated network laboratory to teach advanced networks and security. *SIGCSE Bulletin*, 33(1), Mar. 2001.
- [12] S. Kinnari, A. Suhonen, J. Harju, and K.-J. Riih a. Evaluating a remote access networking laboratory as a learning environment. In *WBED'07: Proceedings of the sixth conference on IASTED International Conference Web-Based Education*, pages 73–79, Chamonix, France, Mar. 2007.
- [13] V. Marozas, R. Jurkonis, and A. Luko evi ius. Development of virtual and remote lab experimentation system for electronics engineering. *Electronics and Electrical Engineering*, (87):41–44, 2008.
- [14] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. In *Proc. of HotNets-I*, Princeton, NJ, Oct. 2002.
- [15] W. F. Tichy. Should computer scientists experiment more? *Computer*, 31(5):32–40, May 1998.
- [16] A. R. White, P. E. McClean, and B. M. Slator. The virtual cell: An interactive, virtual environment for cell biology. In *Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA)*, pages 1442–1443, Seattle, WA, June 1999.
- [17] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation*, pages 255–270, Boston, MA, Dec. 2002. USENIX Association.
- [18] C. Wiseman, K. Wong, T. Wolf, and S. Gorinsky. Operational experience with a virtual networking laboratory. In *Proc. of the 39th SIGCSE Technical Symposium on Computer Science Education*, pages 427–431, Portland, OR, Mar. 2008.
- [19] T. Wolf. Assessing student learning in a virtual laboratory environment. *IEEE Transactions on Education*, 2009.
- [20] K. Wong, T. Wolf, S. Gorinsky, and J. Turner. Teaching experiences with a virtual network laboratory. In *Proc. of the 38th SIGCSE Technical Symposium on Computer Science Education*, pages 481–485, Covington, KY, Mar. 2007.
- [21] T. R. Wyatt, P. Arduino, and E. J. Macari. Assessment of a virtual laboratory for geotechnical engineering education. *ASEE Computers in Education Journal*, 10(2):27–35, Apr. 2000.
- [22] S. E. Yarbrough and R. B. Gilbert. Development, implementation, and preliminary assessment of virtual laboratory. *Journal of Professional Issues in Engineering Education and Practice*, 125(4):147–151, Oct. 1999.