Seattle

Making Value for Society

An Automatic Grading and Feedback System for E-Learning in Information Technology Education

Dr. Peng Li, East Carolina University
Mr. Lee Toderick, East Carolina University

An Automatic Grading and Feedback System for E-Learning in Information Technology Education

1. INTRODUCTION

In the past few years, new, e-learning, virtual hands-on labs have been deployed in the Information and Computer Technology Program at East Carolina University to meet the needs of the growing number of face-to-face and distance education students[6]. Laboratory experiments are useful in teaching students skills needed for employment, and to reinforce theoretical concepts introduced during lectures. However, the students sometimes do not receive timely feedback because it is extremely time-consuming for the instructors to grade the hands-on exercises. The students usually do not have the opportunities to fix the mistakes they made in previous labs due to lack of timely feedback. These challenges were addressed by the creation of an automated grading and feedback system, called BroadReach Extended, for locally designed hands-on exercises in Information Technology (IT) courses. By introducing the automatic grading system, the students receive immediate feedback after submitting the deliverables electronically. Students can learn from the mistakes and resubmit the labs. The students will be able to learn better through "trial and error". Automatic grading scales extremely well, with minimal increase in instructor time.

The idea of automated assessment is not new. Many automatic grading/assessment systems[3,7] have been developed for programming courses since as early as 1960[4]. The history and the recent development in this field were well presented in several systematic literature reviews[1,2,5]. Douce et al.[2] identified three broad generations of test-based assessment systems. Ala-Mutka[1] discussed two generic techniques used by the assessment systems: **dynamic assessment**, based on executing the student code against test data, and **static assessment**, based on collecting information from student code without executing it. BroadReach Extended roughly fits the description of the current (third) generation of assessment tools, which use modern web-based technologies and provide supplementary support for instructors in the form of assessment management and reports[2]. BroadReach Extended uses both dynamic assessment and static assessment. However, the main objects to assess are not student programming codes but deliverables (tasks), defined in hands-on lab manuals.

BroadReach Extended is different from many existing assessment systems in that it is designed for hands-on information technology labs with clear, defined objectives. It is equipped with some unique features such as Virtual Private Network (VPN) and integrity checks. The system is especially suitable for labs which can be performed within a virtualized environment. The environment can be as small as a virtual machine running on students' personal computers or as large as a virtual private cloud hosted on the University campus. In this paper, we will first describe BroadReach Extended for faculty interested in developing their own automated grading solutions. Then we will present the feedback from the students using the system.

2. BROADREACH AND BROADREACH EXTENDED

BroadReach Extended, and its predecessor BroadReach, were developed at East Carolina University (ECU), by Mr. Lee Toderick, the co-author of this paper.

BroadReach was created to provide automated, secure, remote connectivity between distributed nodes (on or off campus) and central server resources located on campus. BroadReach Extended (BRE) was recently added to provide automatic assessment for locally created remote lab experiments to students. This section describes the functions of both.

2.1 BroadReach

Refer to Figure 1. The purpose of BroadReach is to use IPsec to create a secure connection between two remote locations, regardless of the computer network a packet travels across[8]. Unlike some tunneling protocols such as SSH, IPSec passes all traffic through the virtual private network (VPN) tunnel, such as TCP, UDP, or ICMP. The student computer, or a remote virtual environment such as Virtual Computing Lab (VCL), shown in Figure 1, bullet 1, contains one or more virtual machines (VMs), shown in Figure 1, bullet 2. A single VM is used to create one end of the IPSec Tunnel in telecommuter mode with the VPN Gateway, shown in Figure 1, bullet 3. The VPN Gateway is configured in IPsec gateway mode, which protects an internal, private network. In this secure configuration, the VM is logically placed inside the VPN network by IPsec, permitting secure communication between the VM and any VPN network host.
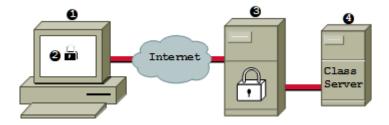

Figure 1. Topology of BroadReach Extended

Because of the complexity of IPsec configuration and the ramifications of misconfiguration, such as connectivity degradation/failure or insecure communication, an automated VPN setup script that was written at ECU is loaded on the VM. The student launches the script, supplies authentication credentials, and the script completes the IPsec configuration process. The script starts the VPN tunnel, and then tests for connectivity to a VPN device. To the student, the VM joins the internal network and becomes another computer on the VPN. Using an automated script allows students new to IT or unfamiliar with IPsec complexity to use the VPN tunnel and complete remote lab experiments securely.

The VPN Gateway is a VM that has multiple custom scripts which comprise an IPsec menu system. The VPN administrator adds and removes a class, and is responsible for overall VPN administration. Instructors manage student access using a script controlling the relevant connection information for each student. Figure 2 shows VPN management activities. Menu options in red are VPN administrator options. Menu options in green are instructor options.

```
VPN Statistics Menu for course: ████████████

Type  Action
c        (c)heck memory- verify memory usage
s        (s)anity check- verify ipsec.conf against the reports files
a        Check class-server (a)ccess-log
f        View IPSec log (f)ile from VPN Gateway
g        View class-server (g)rading error log
l        (l)ist users already assigned
o        List users recently c(o)nnected
v        list user (v)PN Gateway connection status
p        (p)ing an active VPN connection
q        Test class-server's interface to the external network
u        To list (u)sers and ip addresses assigned
V        To show users with a current (V)PN connection

e        (e)xit

Please choose an option:█
```

Figure 2. VPN Gateway VPN Menu Screen for Instructor

Also on the VPN Gateway are scripts that provide instructors with student connection status information, available through both daily email updates and/or real-time, menu driven, selection options. For example, the instructor can choose option o to list the users who have recently connected to the VPN gateway. The connection data may help the instructor identify the students who fall behind or need help.

2.2 BroadReach Extended

BroadReach Extended (BRE) adds a Class Server to BroadReach, shown in Figure 1, bullet 4. The BRE Class Server manages the work of lab grading and feedback, including responding to student-initiated lab grading, sending results to students and faculty, and storing results in log files on the server.

The Class Server is not used by the students to perform hands-on labs. The labs are conducted by the students in virtual machines (VMs) on their personal computers or on virtual environments hosted on campus, as shown in Figure 1, bullets 1 and 2. The students perform tasks (deliverables) specified in the lab manuals. The completed tasks are graded automatically using the grading scripts hosted on the Class Server. It is not uncommon that there are different ways of completing a task. For example, to deny access to an ftp server from an IP address, the students can use IP Tables Firewall or TCP Wrappers. The grading scripts check the end results (deliverables), not how the tasks are completed. A lab may contain as many as 40-50 tasks. The students need to demonstrate whether they have met the learning outcomes by completing these tasks.

The Class Server contains a web server that provides the student with access to lab instructions. When doing a lab, the student first logs in to her/his VM and sets up the VPN connection with the Class Server. Then she/he uses a web browser (e.g. Mozilla Firefox) to access a web site hosted by the Class Server. On the web site, the student selects a class lab link, and a new screen is displayed that provides the lab instructions, a "Submit Lab Work" link, and any lab actions a

student needs to initiate from the Class Server. A sample screen is shown in the top half of Figure 3. When the student finishes the lab on her/his VMs, she/he clicks the "Submit Lab Work" link. This causes a CGI script on the Class Server to send a grading script to the student VM via the VPN. The CGI script executes the grading script. Real-time grading script progress is forwarded to the student's web browser. Upon grading script completion, the CGI script forwards lab results and relevant debug/feedback information to the student web browser, as shown in the bottom half of Figure 3. On the Class Server, summary lab results are appended to the summary log file, which is accessible by the instructor. What is displayed to the student is also retained in a separate file, for use by the instructor if the student requests asynchronous assistance. If the student did not complete the lab successfully, debug/feedback information can be used to fix the issue and the student can resubmit the lab for grade. The entire process is student-centric; that is, the student controls the entire lab experience, beginning with performing the lab experiment and concluding with executing the grading process.

Figure 3. BroadReach Extended Website and Immediate Feedback

Creating the lab experiments and grading scripts is the most difficult and time-consuming element of the BRE system. Lab experiments must be created in such a way as to identify each major task (deliverable) with a grading module that can accurately evaluate student work and provide feedback for incorrect results. Current scripts for Linux hosts are written in BASH (Bourne Again Shell), using built-in and utility commands. Several hours may be required to create a script, and much attention must be paid to the feedback section of each deliverable. The feedback section provides the student with hints as to what could be misconfigured or incorrect, and enables the student to fix a problem and earn a higher grade. The goal of the trial-and-error approach is to help the students learn from the mistakes. Figure 4 displays sample feedback sent by BRE to the student's web browser, showing correct and incorrect deliverables. Extra time is required for the instructors and the student assistants to test the grading scripts to identify and fix grading or feedback errors before the scripts are deployed.
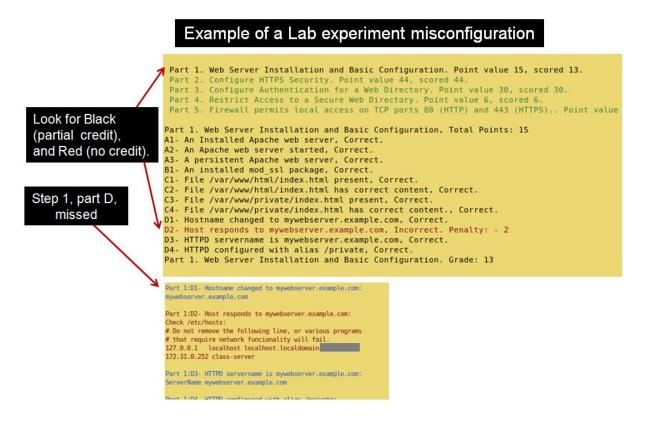

Figure 4. Debugging/Feedback Information by BroadReach Extended

3. USING BROADREACH EXTENDED IN IT COURSES

Using BroadReach Extended, students can submit their lab work electronically, review the feedback on their web browsers immediately, and correct mistakes timely. A graded lab experiment may be resubmitted and re-graded for a higher score. Many students are motivated to find and fix their mistakes. The grading data can also be used by the instructors for assessment purposes. An academic integrity system ensures that only the computer connection assigned to that student is used by that student. The entire system is protected by IPSec, an encryption mechanism securing end-to-end connectivity between the student and the assessment server.

Supported by a University teaching grant, grading scripts for BroadReach Extended were developed and used in fall 2014 for five case studies in ICTN 3540/01: Network Environment III and first four labs in ICTN 4200/01: Intrusion Detection Technologies. The two courses were taught by different instructors.

The case studies in ICTN 3540/01 include

- CS1 - Construction of a Secure Web Server with a Protected Container
- CS2 - iSCSI Initiator w/ iSCSI Target Configuration with RHEL 6
- CS3 - Configure Remote Logging Services with RHEL 6
- CS4 - Implement LVM Storage with RHEL 6
- CS5 - RAID Storage Administration

The first four labs in ICTN 4200/01 are

- Lab 1 - TCP/UDP Ports and Internet Footprinting
- Lab 2 - Linux System Integrity Check using AIDE
- Lab 3 - Host-Based Intrusion Detection System OSSEC
- Lab 4 - Using Packet Sniffing Tools Wireshark and Tcpdump

In ICTN 3540/01, a survey was conducted after each case study. In ICTN 4200/01, a lab survey was administered at the end of each lab and a class survey was administered at the end of the semester. All surveys were anonymous and optional. They were provided online using the Qualtrics survey software. Tables 1-6 show the results from the surveys.

Table 1: ICTN 4200/01 Lab Scores and Submission Attempts

	Number of Students	First Attempt Average Score	Final Average Score	Final Median Score	Average Number of Attempts	Median Number of Attempts
Lab 1	55	36.1	93.5	100	28.2	16
Lab 2	54	49.4	97.1	100	8.6	5
Lab 3	54	61.1	95.9	100	7.6	4
Lab 4	53	31.2	88.3	94	20.6	14

Table 2: ICTN3540/01 Case Study Scores and Submission Attempts

	Number of Students	First Attempt Average Score	Final Average Score	Final Median Score	Average Number of Attempts	Median Number of Attempts
CS1	28	45.2	93.9	100	6	5
CS2	27	46.2	94.6	100	6	5
CS3	25	43.6	95.2	100	4	3
CS4	21	27.5	85.0	100	10	6
CS5	21	44.8	95.0	100	6	6

Table 1 and Table 2 show the scores and the numbers of submission attempts for the labs and the case studies. The first attempt average score is the mean score of the first submission attempts by students. The students can examine the feedback from BRE, fix their mistakes and resubmit their work from grading. As indicated by the average number of attempts and the median number of attempts, many students submitted multiple times after receiving immediate feedback. The students tended to submit more times when the exercise was more challenging and the average score was lower. The final average score is the mean score of the last submission attempts by students. The final average scores are much higher than the first attempt average scores, indicating that the students are able to learn from mistakes and complete most tasks in the hands-on labs eventually. Some students submitted the labs before completing all tasks, resulting in low first attempt scores. At this moment, we only limited the number of submissions in hands-on based, final lab exams. In the future, we can limit the numbers of resubmissions in ordinary labs to discourage mindless resubmissions.

Table 3 and Table 4 show that most respondents in both classes deem that the automated grading system (BroadReach Extended) is easy to use and is effective. It is important to note that the students agree that the immediate feedback help them learn from mistakes. Most respondents agree that the hands-on exercises in the virtual lab environment are as effective as those in physical computer labs.

Table 3: ICTN 4200/01 Lab Surveys
(Strongly Agree = 5; Agree =4; Neutral=3; Disagree=2; Strongly Disagree = 1)

	The automated grading system was easy to use	The immediate feedback helped me learn from mistakes	The hands-on exercises in the virtual lab environment were as effective as those in physical computer labs	Response Rate
Lab 1	4.70	4.70	4.40	10/55
Lab 2	4.79	4.36	4.57	14/54
Lab 3	4.75	4.50	4.75	4/54
Lab 4	5.0	4.80	5.0	5/53

Table 4: ICTN 3540/01 Survey Results from Five Case Studies (N = 82)

	Agree or Strongly Agree (%)	Disagree or Strongly Disagree (%)	Neutral (%)
Case Study Effectiveness.	95.1	3.7	1.2
It was easy to grade and get feedback.	90.2	2.4	7.4
Immediate feedback helps me learn from my mistakes.	91.5	0	8.5
I feel the hands-on exercises in a virtual lab environment were as effective as physical computer labs.	96.3	0	3.7

Results in Table 5 and Table 6 further confirm the students' enthusiasm about using the automatic grading and immediate feedback system BroadReach Extended. As one student summarized: "If I did not get a concept. I know right away and can try something else. The relief of getting the lab correct and knowing right away is a big plus, as well." Some students suggested that the debugging information provided by BRE could be more detailed. However, it is important for the system to keep the balance by providing useful feedback without disclosing too much information or giving away the answers. Too little or too much feedback will not help students develop trouble-shooting skills.

Table 5: ICTN 4200/01 Comments from the End of Class Survey (Response Rate = 9/54)

What are the two things you like most about automated grading and immediate feedback in Labs 1-4?	What are the two things you dislike most about automated grading and immediate feedback in Labs 1-4?
• It was instant grading to see what you missed so you can re-do your mistake until you got it correct and it was easy to use. • Immediate response. • It helped me learn from my mistakes because it allowed me to fix them. It was also nice to know my grades right away, which lead to less stress. • You could easily see what you were not getting correct. • I liked the immediate feedback and the fact that we could re-submit to correct mistakes. • Instantly shows you where you got wrong in a question. • Immediate feedback is the number one thing! If I did not get a concept, I know right away and can try something else. The relief of getting the lab correct and knowing right away is a big plus, as well.	• I didn't dislike anything about the automated grading. • This system limits the lab to close questions. • Nothing. • As I was familiar with the process from having done something similar in Linux classes, I found it straightforward and did not dislike the process at all. • Nothing comes to mind!

Table 6: ICTN 3540/01 Selected Comments from Case Study Surveys

Case Study	What are the two things you like most about automated grading and immediate feedback?	What are the two things you dislike most about automated grading and immediate feedback?
1	The feedback was very helpful in determining what errors were made. The automated grading was fast and I liked the immediate results.	Debug information is somewhat cryptic.
2	I like the breakdown of the which part of the deliverables were not met. and how you get partial credit for some parts.	I would probably like to have more detail in the feedback, maybe even exactly what the output should show.
3	I loved it! Could not have been any better! Fast and efficient.	Needs better descriptions of debug info.

4	CGI scripts are ta bomb. Feedback lets me know exactly what I did wrong.	Feedback could be clearer for missed mistakes. That is all.
5	I love how it is very fast and the feedback is very specific so if you did happen to get something wrong, you are able to know exactly where you went wrong so you can go back and fix the part that you were wrong in.	Can't exactly think of anything, it is great how the feedback and automatic grading works.

4. CONCLUSION AND FUTURE WORK

BroadReach Extended (BRE) provides hands-on, performance-based labs that demonstrate student competency of the subject matter. It enables students to submit deliverables to a central server for immediate grading. The BRE system can be used to monitor student activities on compute nodes (login time, duration, unauthorized contact, etc) and to provide real-time or asynchronous instructor assistance when needed. BRE provides students with multiple trial-and-error opportunities to enhance hands-on and problem-solving skills. The system does not require major changes to the curriculum. The solution is scalable and can be adjusted quickly and deployed in many other hands-on, e-learning courses which are offered on campus or online. The scalability is really important for educational institutions where enrollment is growing faster than the budget and the resource.

After using BroadReach Extended, the students repeatedly express satisfaction with immediate lab experiment grading and the ability to use feedback to correct lab work and resubmit the lab to achieve the maximum grade.

In the future, we would like to expand the BRE project and make the system more usable to a greater number of faculty teaching technical classes with labs. Grading and feedback scripts can be difficult and tedious, and may not be possible for all faculty to create. A future BRE upgrade is intended to include a menu-driven module for less experienced faculty, and an advanced module that accepts complex programming statements for experienced faculty.

In addition to the two courses in this paper, BoardReach Extended has been deployed in other courses inside and outside East Carolina University. Data collection is still ongoing and the results will be reported later.

REFERENCES

[1] Ala-Mutka, K. M. (2005). A survey of automated assessment approaches for programming assignments. *Computer Science Education*, *15*(2), 83-102.

[2] Douce, C., Livingstone, D., & Orwell, J. (2005). Automatic test-based assessment of programming: A review. *Journal on Educational Resources in Computing (JERIC)*, *5*(3), 4.

[3] Edwards, S. H. (2003). Improving student performance by evaluating how well students test their own programs. *Journal on Educational Resources in Computing (JERIC)*, *3*(3), 1.

[4] Hollingsworth, J. (1960). Automatic graders for programming classes. *Communications of the ACM*, *3*(10), 528-529.

[5] Ihantola, P., Ahoniemi, T., Karavirta, V., & Seppälä, O. (2010, October). Review of recent systems for automatic assessment of programming assignments. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research* (pp. 86-93). ACM.

[6] Li, P., & Toderick, L. W. (2010, October). Cloud in cloud: approaches and implementations. In *Proceedings of the 2010 ACM Conference on Information Technology Education* (pp. 105-110). ACM.

[7] Saikkonen, R., Malmi, L., & Korhonen, A. (2001, June). Fully automatic assessment of programming exercises. In *ACM SIGCSE Bulletin* (Vol. 33, No. 3, pp. 133-136). ACM.

[8] Toderick, L. W., & Lunsford II, P. J. (2007, October). Using VPN technology to remove physical barriers in linux lab experiments. In *Proceedings of the 8th ACM SIGITE Conference on Information Technology Education* (pp. 113-118). ACM.