Work in Progress – Re-examining Closed Laboratories in Computer Science Courses

Thomas B. Horton, Ruth E. Anderson, and Christopher W. Milner
University of Virginia, Department of Computer Science,
Charlottesville, VA 22904
horton@cs.virginia.edu, rea9x@cs.virginia.edu, cmilner@cs.virginia.edu

***Abstract* – We present early results of a project that ultimately seeks to define how best to design, implement and deliver the next generation of laboratories for courses in computer science. In response to *Computing Curriculum 1991* many computing programs adopted closed laboratories, especially for early courses. But advances in computing since 1991 raise questions about why closed laboratories should be used. The project has begun with a re-examination of the goals and strengths of the closed laboratory model. We have defined a classification of the types of activities students carry out in closed laboratories, along with set of rationales for why an activity might be included in a closed laboratory. These results were used to modify some closed laboratories in our CS1 course. We believe that the more complete understanding that our analysis of closed laboratories provides will help us design more effective practicum-oriented experiences for students in computing courses.**

Index Terms – Closed-laboratories, Computer science education, Laboratories

The high-level goal of this project is to develop educational support resources that allow computing educators to design, implement and deliver the next generation of laboratories for computer science education. Closed laboratories have been adopted by many universities since the publication of *Computing Curriculum 1991* [2]. Many faculty and students believe they are useful, and there are published results that indicate they lead to improved performance on exams [1]. But the kinds of educational activities that take place in closed laboratories vary widely, and we believe the computing educational community could benefit from a better understanding of why and how closed laboratories should be used in a degree program.

Our work to this point has identified a *classification of student activities* that take place in closed laboratories, as well as a *set of rationales* for placing these into a closed-laboratory setting. Understanding this classification and these rationales is a start, but our current works seeks to address some important questions that this understanding leads to:

- If closed laboratories are educationally effective, which of the identified activities are the main reasons for this effectiveness?
- Which of the identified activities and rationales that characterize closed laboratories have particular pedagogical interest for computing educators?
- Are there alternative ways to satisfy the goals for closed laboratories that are more resource effective? (I.e. that can be developed and delivered more easily, with fewer instructor resources, with fewer hours in the laboratory, etc.) Given that developing and offering closed laboratories with courses has a high cost in resources, can we divide activities between closed laboratories and their alternatives (open laboratories, homework, etc.) in order to use closed laboratories most effectively and efficiently?

The immediate goals of our current work are to create laboratory development guides that document:

- guidance for creating lab-based educational exercises,
- best practices for developing and delivering materials for course laboratories,
- information on how to evaluate the educational effectiveness of laboratory exercises,
- advice on developing and delivering such materials in a cost-effective manner.

Thus our goal is not to define the *content* of "next generation" laboratories in computing, but to define how educators can best use laboratories in a computing curriculum. This work is based on a re-examination of the closed laboratory concept based on "first principles." This includes a review of the rationale for laboratories expressed in *Computing Curriculum 1991* [2] in light of changes in technology and education since that time.

Our project seeks to identify what kind of activities are best suited for inclusion in a closed-laboratory class setting, allowing instructors to make the best use of a closed laboratory. Thus two important objectives are *development cost reduction* and *operational cost reduction*. We recognize that closed-laboratories require significant time for instructors to develop materials and to deliver them in a laboratory session. Students also desire that lab sessions not waste their time doing something they could do at a time and place of their own choosing. Thus our work will produce guidance that will help instructors use closed laboratories in their programs in a way that maximizes the development and operational cost in terms of educational effectiveness.

CLASSIFICATION OF ACTIVITIES

We have identified a classification of types of activities that are often included in closed laboratory courses in computing. Students may:

- Type 1 - Carry out a guided sequence of activities. (Leads to an overall understanding through smaller steps.)
- Type 2 - Gain proficiency using a software or hardware system that is difficult for the instructor to demonstrate during lecture and/or is difficult for the students to learn without a hands-on learning opportunity. (This may be combined with a Type 6 activity.)
- Type 3 - Demonstrate or present a work product that has been prepared beforehand.
- Type 4 - Create a work product in a fixed amount of time.
- Type 5 - Take a quiz or exam (i.e. a more formal assessment of student performance).
- Type 6 - Listen to a lecture or observe a demo, or participate in a discussion or question/answer session. (This is similar to a discussion section or a recitation section.) A demo might be a video or some kind of digital movie etc. viewed on a computer.
- Type 7 - Carry out an experiment (i.e. measure something, e.g. timing an algorithm)
- Type 8 - Collaborate with other students in any of the above types of activity.

This list of types of activities was based on analyzing what activities we carry out in all of our closed laboratories. We also examined laboratory manuals for a number of current textbooks. We also reviewed on-line support materials for an introductory Java textbook that were noted as being extensive and popular with instructors by representatives from a publisher. This list of lab activities has also been reviewed by peers at our university who are not associated with our project.

RATIONALE FOR ACTIVITIES

In addition, we asked ourselves why any student learning activity should be located in a closed laboratory. This led to the following list of rationales for locating activities in this environment:

1. *Compliance.* We can know that they have carried out one or more particular educational activities, and possibly record that they have done it.
2. *Demonstrate a particular understanding or competence.* Sometimes it is important to demonstrate something to a person (e.g. a TA), or to demonstrate that a task can be done in a limited time period. A closed lab facilitates this.
3. *Get assistance from instructional staff.* Carrying out an educational activity may go better if there are instructors or TAs on hand to assist or advise.
4. *Interact with other students.* Students may collaborate in groups or pairs. They may observe others present their work or results. They may help, support, compete with, or encourage each other as they carry out some educational activity.
5. *Control how they carry out an educational activity.* If there is more than one way to complete an activity, and the choice of solution or the process is important, than TAs may guide or lead students through an activity.
6. *Teach material.* A closed lab may include a short lecture or lesson given by the TA.
7. *Development and practice of skills.* Educational activities may focus on the practice of practical skills, such as using a programming tool, computing environment or language construct. This may involve repetitions, and the TAs or other students might assist. These more experienced people may point out techniques or short-cuts that are very relevant to a student's immediate needs.
8. *Provide access to computing resources.* Students might come to a closed laboratory session in order to access hardware or software that is not available to students at home or in other labs. These might include computers, cameras, robots, or software products that only run in the labs.
9. *Provide a satisfying, interesting or fun shared, common experience.*
10. *Provide a controlled experimental computing environment* for certain specialized kinds of computing experiments involving measurement, performance, etc.

This analysis leads to some general observations. Many things have not changed since the use of closed laboratories was emphasized in the early 1990s. Students still need to gain proficiency with systems (activity type 1) or carry out an experiment (activity type 7), and closed laboratories are a natural way of making sure they have completed the activity (rationale 1) and learned this skill (rationale 2), and of giving them assistance (rationale 3) as they learn. However, giving them access to quality computing resources (rationale 8) is often less important now because many students have excellent, convenient access to networked machines at home. Another change is that educators are now more interested in team and other inter-personal interactions among students (rationales 4 and 9); pair-programming is an example of this, as well as the use of contests in introductory courses.

We will give examples of how we have made use of these results to revise existing laboratory experiences in our CS courses, and provide assessment information on their effectiveness.

ACKNOWLEDGMENT

Support for this work was provided by the National Science Foundation's Course, Curriculum and Laboratory Improvement program under grant DUE-0127452.

REFERENCES

[1] Thweatt, M. "CS1 Closed Lab vs. Open Lab Experiment." Proceedings of the twenty-fifth SIGCSE symposium on Computer science education, 1994, pp. 80-82.

[2] Tucker, Allen (editor), et al, "Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force", ACM Press Order Number 201910, IEEE CS Press Order Number 2220.