

Virtual Computer Science Laboratory

An OOPSLA 2000 Design Fest Exercise

Joseph Bergin
Pace University
jbergin@pace.edu

Abstract

In computer science education, many universities find it difficult to have closed labs because of lack of space or scheduling difficulties. The net makes it possible to conceive of a "virtual" computer science lab in which students could work collaboratively over the net on relatively small problems. Pairs of students and small teams would work asynchronously, but over a short time frame on problems set by the instructor and the system would make it possible for the instructor to "look over the shoulders" of the students and provide help. It might even be possible to provide an artificially intelligent lab instructor that could provide advice.

The software envisioned here would provide such an environment. Different people would have different roles within the system. The main roles are instructor and student. Each person would interact with an appropriate client. The system would make it possible for a group of students to be broken into teams and for each team to work synchronously and asynchronously on a problem set by the instructor. The system will permit common document edit and markup by team members. It will be modular so that different modules may be responsible for different document types. The system will also provide various communication facilities (bulletin board, email, chat,...) so that teams may work effectively. The instructor must be able to connect to any team at any time to review progress and make suggestions.

Background

In computer science education there are traditionally two kinds of lab exercises. Closed labs traditionally bring all of the students in a course together in one place with a lab instructor. Small assignments are given and students work in teams to solve them. The meeting place is equipped with appropriate equipment, usually stand-alone or networked computers. This is similar to what one normally thinks of as a physics lab or a chemistry lab. Students have a few minutes to a few hours to complete the exercise. These exercises often form a small part of a student's grade for the course. Collaboration is usually encouraged. The instructor is always available to answer questions and provide appropriate hints. Such labs require pre scheduling of all participants as well as the room in which the class will meet. The room typically contains expensive equipment. Either the scheduling or the cost is a serious impediment to using this methodology, though it has proven effective. It is especially valuable in the first courses in a curriculum.

Open labs, on the other hand, are just assignments given to students, often singly and less often in teams. Students typically have a few days to a few weeks to complete the exercise. They work on their own,

either on personally owned computers or on university owned machines in general purpose machine banks. These exercises often form a fairly large part of a student's grade. Collaboration is often discouraged. The instructor may not be available for long periods as the students work on the exercise. This methodology is more valuable in upper-level courses, but is less time and money intensive for the university, so tends to be used as a default even when closed labs would be preferred.

A Proposal

This design exercise attempts to begin the development of a third alternative, the virtual lab, that tries to combine the sharing and supervision of the closed labs with the cost and scheduling advantages of open labs.

A virtual lab is a time set aside for communication over the net by students and a lab instructor (who may well be the course instructor). The time may be short or long, and the communication may be strictly synchronous or partly asynchronous. The software designed here would be the communication medium and would provide appropriate book keeping support as well, such as guaranteeing that time restrictions were met.

- A lab session is initiated by the lab instructor who assigns students to teams (of 1 or more) and poses one or more problems for solution.
- The instructor also configures the system as appropriate with modules representing different roles, document types, etc. if available.
- Students connecting to the system will learn who their lab partners are and the nature of the problem.
- Students will be automatically connected to their partners (upon the partners connection to the system) and they may communicate with each other immediately using a feature like a chat room.
- A team of students may create and jointly edit various documents in various formats (html, text,...) as part of the exercise. The system will maintain these documents and permit their markup.
- Students may also pose questions of the instructor at any time.
- The instructor may respond to a group and may also broadcast answers (as well as incoming questions) to the entire class.
- If a student has not connected at the beginning of the period, they will have access to a transcript of all communications with their group.
- The system will also log all communication of all groups so that the members of the group may review their work and the instructor may review all work.
- The students need to be able to submit final work as a group to the instructor.
- The system must be flexible so that additional roles in addition to instructor and student may be added.
- The student learning time for the system should be short.
- The system must not depend on particular hardware on either the client or server end, though it might depend on certain performance capabilities.
- The server will need high availability. It will need to be able to handle one course at a time, but must be capable of being spawned for several courses.

Various Extensions and Commentary

It would be advantageous if a group formed by the instructor could form subgroups for subtasks and have the system treat subgroups just as it treats groups, with it additionally being required that a

subgroup needs to be able to submit its results to its group.

In certain circumstances an instructor might want to restrict communication between groups and in others to make such communication easy.

The problems posed by the instructor may be programs to write, designs to create and document, or other problems. It is not possible to predict all of the ways that such a system might be used. It would be advantageous to have the system built with a component architecture so that specific support for different kinds of problems might be added in the future. Having a compiler built into the system, for example, is not a firm requirement, but might greatly enhance the system for certain kinds of problems. The same would be true of a design documentation tool. It is expected, however, that students would have a full range of tools available outside this system on which they might prepare materials. These need to be incorporated into the discussions, however, and might consist of text, graphics, and even multimedia. The best system would permit markup of all such submitted materials. Markup could be done by other team members and by the instructor.

The system might be component based at many levels, permitting, for example, an instructor to plug in an artificially intelligent tutor to monitor the groups in the human instructor's absence.

The instructor must be able to quickly and simply join a group in progress and review the history of any group.

The most useful system would permit a group to work partly asynchronously, with different members connected at different times. They would need to be able to review past work, make their own contributions, and then leave to return later. This way a "lab period" might extend over a period of a few days, rather than just a few hours. The system would be able to enforce a firm deadline for completion, though this is not always desirable.

The most useful system would also track contributions by participant. This would require appropriate notice of all participants, of course, and might also be a feature that could be disabled or enabled for a particular lab.

The most useful system would permit easy file exchange within a group and between groups and instructors.

There might be several instructors and they might want to divide up the groups among themselves.

The Lab Environment might actually consist of a set of tools, rather than a single tool. Cut and paste from/to other tools would be essential, however.

A Use Case (not really a use case--more like a use scenario)

Professor Jones has a class of 30 students in CS1. She has previously entered her roster containing names and email addresses of her students. She decides to have a virtual lab that will run from Monday noon until Wednesday noon with the student randomly assigned (by the system) to groups of three. The lab will be "Exploring Selection Primitives" and will consist of three parts. Each part will be automatically distributed to a group when the previous part is submitted by that group. She logs into the system on Friday, sets up the groups, enters the document URLs for the three parts of the lab and then enters the announcement of the beginning of the lab. This announcement is then delivered to each student, along

with the names and email addresses of each group member. She also chooses the option of having questions to the instructor forwarded to her email address.

On Monday at 3 PM student Smith logs in and gets the first task in the lab. He is the first to log in from his group so he enters some preliminary notes on this part. While he is typing, student Nguyen logs in and is a member of the same group and reads what Smith is doing and starts a chat. Between them they outline this part and log off, promising to meet again at 7 PM and sending a note to the third member that things have begun. At 7 PM (or so) all three meet in the virtual lab to finalize the first part and submit it. They notice that the professor has commented on their outline and so they respond to this and submit part one. Part 2 is immediately delivered and they work until 8 on this part. Smith promises to do a bit of research on one of the elements and report back to the group by 10 am the next day. He leaves while the others work a bit longer and then leave a question for the instructor and quit.

The instructor, upon logging in at 7 am notices several queries from different groups. They fall into three broad categories, so she writes a reply and broadcasts the questions and her answers to all groups, though she could have sent individual responses. She then reviews the ten transcripts and comments on four of them.

...

The lab closes at noon on Wednesday and solutions are posted to all groups.

Code Fest Suggestions

The above is too big to consider for a week's work at Code Fest. Consider the following for implementation.

The most important requirement is flexibility and extensibility. The implementation must not put restrictions on later enhancements. The key dimensions of extensibility are people's roles (instructor, student,...), document types (html, plain text,...), and communication methods (chat, email, ...).

The server must be able to communicate with clients representing different roles and must be able to manage communication between teams. It must also manage a document repository and serve documents to the teams. The server must also manage time so that lab sessions can have a deadline.

Initially the system must support instructor and student roles.

- The instructor role (perhaps a separate client program) must be able to create class rosters, create teams from the class roster, and perform all functions of the student role (markup, ...). This role must additionally be able to communicate with the class as a whole.
- The student role must be able to communicate with team members and with the instructor. It must be able to create and modify documents jointly with other team members and ultimately submit them to the instructor.

Chat and email are sufficient for the initial implementation. Chats must be logged and retrievable by the teams and the instructor.

Text and html documents are sufficient for the initial implementation. (Actually, any two different

document types is OK.)

Note that if a standard browser hosts the clients, then its facilities may be taken advantage of wherever possible. Thus email, for example, might be very easy to provide. There should, however, be no restriction as to which browser is used. Other assumptions about the client hardware and software environment should not be made.

Last updated: July 22, 2000

For Peter Smith, peter@petersmith.net