

2005-2007 LITRE Grant Proposal

1. Project Title:

Learning with In-class Technology (LIT): The 15-Minute Learning Module Approach

2. Project Coordinator:

First Name: George

Last Name: Rouskas

Campus Address: Department of Computer Science, Box 7534

Email Address: rouskas@csc.ncsu.edu

Campus Phone: 515-3860

Unity ID: rouskas

3. Other Participants:

Ms. Carol Miller (co-PI)

Department of Computer Science

miller@csc.ncsu.edu

515-2042

4. College or Unit:

Engineering

5. Department:

Computer Science

6. Project Description:

We propose to develop a general framework for using technology in the classroom, and apply it to the CSC 216/316 course sequence by implementing a set of learning modules to demonstrate fundamental programming concepts. Students with laptops will be able to download and use the modules during the lecture period to gain hands-on experience with the corresponding concepts.

CSC 216 (Programming Concepts) and CSC 316 (Data Structures) are core Computer Science courses, and are required courses for Computer Engineering students. Approximately 750 students currently enroll in CSC 216/316 per school year. The PIs have recently taught, and will continue to teach, laptop sections of CSC 216 (Miller, Fall 2004, Spring/Fall 2005 and Spring 2006) and CSC 316 (Rouskas, Spring 2005 and Spring 2006) using technology in the classroom. These sections were sponsored by the College of Engineering (CoE) Laptop Initiative program which aims to enhance student learning by introducing technology into the classroom experience. Each student and the instructor used laptops during each class meeting.

Given that the vast majority (well over 90%) of incoming CoE freshmen bring their own computer (typically a laptop) to campus, it is envisioned that in the near future all sections of CSC 216 and CSC 316 will eventually transition to the laptop format.

Based on student surveys of the laptop sections conducted by Dr. Rebecca Brent, students were receptive of the new format; her report to one of the PIs (Rouskas) at the end of the Spring 2005 semester states: "Looking over the data, I think there was a very positive response to the technology. Most students seemed to feel that it helped their learning." Similar observations applied to the laptop section of CSC 216. These reassuring findings encouraged us to join our efforts in seeking a LITRE grant to further enhance the student experience in CSC 216 and CSC 316.

While isolated efforts by individual instructors to introduce technology in relatively small, separate sections of a class may yield good results, extending this concept to all sections of a sequence of core courses (such as CSC 216/316), including sections with a large number of students, is an ambitious and challenging task. When implementing such a transition to technology-based instruction on a broader basis, one has to pay attention to the impact on courses that follow the laptop course, as well as on the curriculum overall. Because the CSC 216/316 course sequence provides students with the fundamental concepts of programming, data structures, and algorithms, it plays a central role in the Computer Science and Computer Engineering curriculum. Therefore, this course sequence is an ideal candidate for introducing students to technology-based instruction that complements traditional class lectures.

We seek LITRE funding to design, implement, and incorporate in our CSC 216/316 courses a set of learning modules which students will work on during class. The PIs will be responsible for designing the learning modules and integrating them in their lectures, and they will supervise students to implement and thoroughly test the modules and an associated web-based tool prior to using them in class, and to help with the project assessment plan. The learning modules will be designed to complement the lecture material and reinforce important concepts (refer to "Project Objectives" below).

Developing learning environments that incorporate technology in the classroom is a challenging task. By adapting learning materials to use the technology, the instructor can provide a pedagogy utilizing more learning styles and provide an interactive learning experience for each student.

7. Project Objectives:

Our long term goal in carrying out this project is to develop a general framework for using technology in the classroom, and implement a

specific instance of it for the CSC 216/316 course sequence, which will benefit both instructors and students. For instructors, we hope to raise awareness of how technology can be used in the classroom, support them in developing a new style of classroom learning environment, and provide them the means to move beyond (and "unlearn") the one-way communication typical of most college lectures.

For students, we aim to provide a personalized learning experience within the classroom, improve student confidence with new learning, and convey to learners exactly what needs to be mastered. We will achieve these goals by providing a well-defined and organized plan for productive use of the in-the-classroom-technology, while at the same time decreasing (or even eliminating) the unproductive use of in-the-classroom-technology (e.g., web browsing, IM, email reading) that hampers similar efforts lacking a concrete pedagogy plan.

This project proposes organizing the classroom experience into modules with each module providing a new abstraction, concrete connections to explain the relevance of the material and an exercise for the student. The exercise would make the learner an active learner, stimulating him/her to apply concepts and explore the if/why/how/what of the topic. The results of the exercise would provide instructor feedback to guide the next stage of learning. The modules would provide short (10-15 minute) cycles to be repeated for many/most of the lecture periods. This is the model that has been used by the PIs for the laptop sections of CSC 216 and CSC 316 for the past and current semesters.

This project will provide a framework for developing a set of modules for CSC 216/316, as well as 20 written modules for these two courses (10 modules per course).

The framework will be an interactive web-based tool to assist the instructor in writing and organizing a set of modules. The tool will also provide easy archiving and retrieval of the modules. Each module will contain:

1. a subject/key for easy indexing and retrieval;
2. a section on the abstraction;
3. a section discussing the concrete relevance; and
4. an exercise to be downloaded and used to apply the knowledge of the module.

Students would receive the entire module or portion of the module in class and return the completed exercise to the instructor, all electronically.

Below is a sample framework containing a module.

*****Sample Module Begin*****

Subject:

Binary Search

Abstract:

A binary search of an array-based list is easy and efficient. The array is searched repeatedly by dividing the search interval in half. The search begins with an interval covering the whole array. The item in the middle of the interval is inspected and if it matches the key, the search is complete. If the value of the search key is less than the item in the middle of the interval, the search is narrowed to the interval in the lower half. Otherwise, the search is narrowed to the upper half. The search is repeated until the value is found or the interval is empty.

Concrete Example:

Consider a large city phone book (electronic or paper) that has 1,000,000 unique entries. Using a binary search for a specific name involves inspecting the name in the center of the list. If it matches the name, the search is complete. If the name is less than the name at the middle of the list, repeat the process using the first 500,000 names. If the name is greater, repeat the process using the last 500,000 names. Continue.

Exercise:

Calculate the MAXIMUM number of names that would need to be inspected to find a specific name in the phone book.

What is the MAXIMUM for 2,000,000 names?

Complete the following Java code to implement a binary search of an array of n names. Do the exercise iteratively.

```
public class BinarySearch {

    private String [ ] names; //array of names

    private int numNames; // number of names in the array

    public BinarySearch ( int n) {
        // assume code to populate the array with n names -- to be provided
    }

    /**
     * Search an ordered array for a key
     * @param key of search
     * @return index of key in array or -1 if not found
     */
    public int binarySearch ( String key) {

    }

}
```

}

Discuss 2 cases for this code and explain why each is a good test case.

- 1.
- 2.

*****Sample Module End*****

The framework will be developed as part of this proposal and applied to CSC 216/316. It is intended that this framework could be readily adapted to other courses in Computer Science and other disciplines.

Twenty modules will be developed by the PIs for the CSC216/316 courses. These modules will be used in the 2006-2007 academic year, critiqued and archived for use by all instructors in CSC 216/316.

The exercise component of each module is specific to the concept/abstraction, and the course discipline. For the CSC 216/316 courses that involve programming, the exercise will include the following three parts (refer also to sample module above):

- (a) A software component (Java code) which manipulates the input or implements visualization and other auxiliary functions. This piece of code may not be directly related to the concept emphasized in the module, and may be time consuming to implement and debug. Therefore, it will be provided to the students in advance. In the sample module, the code to read a file with n names and populate an array is an example of such code.
- (b) Code which is directly related to the concept to be reinforced. This piece of code will not be provided in advance; instead, students will have to implement this code in class. For instance, students may be asked to write the code for binary search, as in the above sample.
- (c) Input data, typically in a separate file, that will be used by the software; e.g., a file with n names to fill the array to be searched.

The PIs will jointly design the modules for both classes. Specifically, the following tasks will be carried out:

Task 1. Identify the important abstractions/concepts to be emphasized.

This is a critical task of the whole project, and the PIs will draw on their extensive experience in teaching CSC 216 and CSC 316, as well as their experience with laptop sections of these courses. It is also important to coordinate the modules of CSC 216 and CSC 316 to ensure that: all important concepts/abstractions are covered; the difficulty of the module exercises progressively increases throughout the semester, and from CSC 216 to CSC 316; the overlap between modules across the two courses is minimal; and whenever it is necessary to cover the same concept in both courses (e.g., recursion), the corresponding modules in CSC 316 are at a higher level and involve more complex exercises than those in CSC 216.

Task 2. Determine appropriate in-class exercises for each concept identified in Task 1. We will design these assignments so that they are sufficiently compact to complete within a short time (e.g., fifteen minutes or less) yet powerful enough to convey the important ideas. Whenever appropriate, we will use animation or other visualization tools to demonstrate the intermediate steps and the final result of the process. For certain concepts, we will develop a number of alternative assignments of the same or different difficulty level in order to use in different semesters or different courses, or in order to determine which one is better in conveying the underlying concept.

Task 3. Supervise a student who will implement, debug, and test the web-based framework tool and the twenty modules before incorporating them in our classes.

Task 4. Integrate the modules in the class lectures.

Task 5. Supervise a student to help in carrying out the assessment plan. Specifically, the student will be involved in generating questions for student surveys, collecting the data, and performing quantitative analysis.

During the last two tasks, the PIs will also consult with each other to identify challenges and areas needing improvement, offer suggestions of techniques that work well, and exchange ideas on refining the programming assignments to improve student experience.

We note that the PIs already have experience with developing compact in-class programming assignments for their courses. In particular, each PI received \$2500 from the CoE in Spring 2005 which they used to hire a student who developed a small set of in-class exercises. While this development effort was a good learning experience for the instructors, the process of introducing technology in the classroom on an as-you-go basis has two drawbacks. First, it is ad-hoc in nature, as there is little time for careful design and consideration of the impact of the programming assignments on the students' overall learning process. Second, it is difficult to carefully test and debug the code to ensure that it runs correctly and avoid incompatibility issues that cause delays during the limited lecture time and distract students from the task at hand. Therefore, the PIs plan to implement this project in three phases:

1. a design phase (Tasks 1 and 2, involving effort by the PIs),
2. an implementation phase (Task 2, involving effort by the PIs and a student), and
3. an application and assessment phase (Tasks 4 and 5, involving effort by the PIs and a student); please also refer to the timetable below.

8. Estimated number of students affected:

In the short term, the work on this grant will support students enrolled in the laptop sections of CSC 216/316; typical enrollment numbers are between 30-50 students per section per semester offered.

In the medium to longer term (2-5 years), we expect that most or all sections of CSC 216/316 will transition to a laptop format. Consequently, our work will eventually support all students in these two courses (approximately 750 students per school year).

We also envision that if successful, our framework could provide a model for other courses in Computer Science and other disciplines, extending its impact well beyond CSC 216/316.

9. Outcomes of the Project:

1. Students will acquire core concepts in programming and will apply Java code to illustrate core concepts in the discipline. This outcome relates to LITRE goal d) performance in the discipline.
 2. Students will be able to apply data structures and algorithms to solve a wide range of realistic problems. This outcome relates to LITRE goal a) problem solving, and d) performance in the discipline.
 3. Students will evaluate alternative approaches and assess their strengths and weaknesses with respect to given applications. This outcome relates to LITRE goal a) problem solving, and d) performance in the discipline.
 4. Students will have a positive attitude towards programming by using computers in the classroom.
-

10. Project impact on NCSU:

The PIs propose to design a general framework for using technology in the classroom, and implement and evaluate a specific instance of it for the CSC 216/316 course sequence. This framework can support a pedagogy utilizing more learning styles than traditional one-way lectures, by providing an interactive learning experience for each student.

The proposed learning framework can be readily extended/adapted to a wide range of courses. As such, it may provide a viable and effective model for accelerating the adoption of technology in the classroom across disciplines and has the potential to impact student learning at NCSU.

11. Project Assessment Plan:

1. Special questions at mid-term and final exams will be developed to assess core concepts, ability to apply code, and ability to problem solve. By grading the responses, we will be able to assess outcomes 1 and 2. Where possible these data will be compared to sections not using laptops in class.
2. A selection of student work will also be collected as students implement the code in class and solve problems, say three times over the semester. This data will be used to establish the extent to which

students are able to implement code and solve problems, that is outcomes 1 and 2.

3. Twice during the semester, students will be asked to turn in a class "exit slip" by writing down one example of how they had used problem solving in class. This can be done on an index card. This data will be analyzed qualitatively to identify particular instances of problem solving, and also any common themes. This specifically addresses outcome 2.

4. Students will be asked to write down their responses (in pairs or groups) as they evaluate alternative approaches and assess their strengths and weaknesses with respect to given applications. These will be collected and analyzed for correct and incorrect suggestions from students. This analysis will provide information about outcome 3.

5. Student attitudes to the course will be assessed through surveys, such as the one implemented by Dr. Rebecca Brent in Spring 2005.

12. Staffing and Support:

The PIs (Rouskas and Miller) request \$5,000 each (a total of \$10,000) for summer support in fiscal year 2005-2006 for the design phase of the project (Tasks 1 and 2 identified under "Project Objectives"). From May 15, 2006, to June 30, 2006, the PIs will jointly identify the abstractions/concepts to cover in CSC 216/316, will write specifications for the web-based tool, and will determine appropriate in-class programming assignments for each of the twenty learning modules to be implemented.

The PIs also request \$10,000 in fiscal year 2006-2007 to fund two students who will implement the web-based tool and the twenty modules, and will help with the assessment plan.

The PIs will dedicate their own time and effort to integrate the proposed modules in their courses during the 2006-2007 academic year.

13. Financial Support Requested:

EPA salary total: \$10,000

SPA salary total:

Other salary: \$10,000

Equipment:

Cost associated with assessment:

Other financial support requested:

Total Funds requested: \$20,000

Additional Explanation of how funds will be used:

14. Funding Breakdown:

Total funding requested for fiscal year 2005-2006: \$10,000

Total funding requested for fiscal year 2006-2007: \$10,000

15. Staff Support and/or Technical Support Requested:

16. Timetable for Implementation:

May 15, 2006 - June 30 2006: Design phase.

The PIs will design the twenty class modules by carrying out Tasks 1 and 2.

July - December 2006: Implementation phase.

The PIs will supervise a student who will implement and test web-based tool and the twenty modules (Task 3).

Fall 2006, Spring 2007: Application and Assessment phase.

The PIs will incorporate the programming assignments in their lectures, will identify areas of improvement, and will supervise a student who will carry out the assessment plan (Tasks 4 and 5).

17. Human Subjects Protection:

If your proposal project involves research using human subjects, you will need approval from the Institutional Review Board for the Protection of Human Subjects in Research (IRB) prior to final approval. IRB information is available at <http://www.ncsu.edu/sparcs/irb>



18. Proposal Release:

By submitting this proposal the applicant grants the LITRE Advisory Board permission to make this proposal available as an example for future grant applicants. All personal information will be removed if this proposal is used as an example.

