Pargas, Roy

Development of a Targeted Tablet PC Software Development Course

Contact Information

Department of Computer Science
204 McAdams Hall
Clemson University
Clemson, SC 29634-0974
864.656.5855 (office)
864.656.4771 (cell)
pargas@clemson.edu
http://www.cs.clemson.edu/~pargas

Biography

Roy P. Pargas (Ph.D. Computer Science, University of North Carolina, Chapel Hill) is an Associate Professor of Computer Science at Clemson University. One of his interests is in developing tools to support the instructor teaching courses in technology-enhanced classrooms, i.e., in which all students arrive with laptop or tablet computers with access to the Internet. Two of his current projects are (1) *MessageGrid*, a web-based tool designed to support interactivity in a technology-rich classroom, and (2) developing tablet PC software that demonstrate and animate concepts in courses like CS4 (Algorithms and Data Structures) and CS3 (Computer Organization).

Tablet Project Abstract

New technologies can excite the imagination. This is true not just with a finished product but equally so during the *development* of a product. We propose to capture and infuse students with this excitement through a new course entitled *CPSC 481/681: Tablet PC Software Development*, designed for senior undergraduate / first-year graduate computer science students and offered by the Clemson University Computer Science Department.

This course achieves two interrelated goals. (1) The course teaches both undergraduate and graduate students the skills necessary to develop exciting and useful tablet PC software. (2) The software developed will provide *new* instructional material targeting an existing laptop-enhanced course being taught at Clemson University. In a laptop-enhanced course, each student comes to class with a laptop equipped with wireless access to the Internet. These laptop-enhanced courses are being taught by instructors (from many different disciplines) who are experimenting with novel pedagogical techniques in an attempt to use technology, i.e., laptops or tablet PCs, to deliver course content more effectively. The tablet PC software produced by *CPSC 481/681* students will supplement the material currently being used in these courses.

For example, in Fall 2005, we will target CS4, that is, CPSC 481/681 students will develop tablet PC software that will assist the CS4 instructor teach data structure concepts such as stacks, queues, lists, trees, and graphs. Associated algorithms will demonstrate tree traversal, insertion into and deletion from binary search trees, AVL trees, splay trees, B-trees, binary heaps, leftist and skew heaps, Dijkstra's algorithm for shortest paths, Prim and Kruskal's algorithm for minimum spanning trees, depth- and breadth-first graph searches, and message encoding and decoding using Huffman trees.

Work along these lines has already begun. Current progress includes tablet PC programs for drawing linked lists and trees and algorithms demonstrating inorder, preorder, and postorder tree traversals. A user, for example, can draw an arbitrary binary tree, label the nodes with numeric or alphabetic values, click a button, and observe the nodes of the tree being visited in inorder (Figure 1 shows the final state after an animated tablet PC program demonstrates the order in which nodes are visited). We have tree applications that demonstrate how random values specified by the data structures student using the tablet stylus are inserted into a binary search tree, an AVL tree, or a red-black tree. The plan is to use these examples to demonstrate the CPSC 481/681 how tablet PC programs are developed. After learning the basics, students will develop their own tablet PC programs for the data structures and algorithms they select from the project list.

The result, at the end of the semester, is a collection of tablet PC algorithms that demonstrate a variety of data structures and their behaviors. The benefit to a CS4 instructor is substantial. He or she can, for example, use the tablet PC algorithms to demonstrate how to insert nodes into an initially empty AVL tree and show how the tree self-balances. Or the instructor can draw a graph and compare and contrast Prim's and Kruskal's algorithms for finding minimum spanning trees. The simpler algorithms, such as tree traversal, may also be used in CS1 and CS2.

The set of algorithms is particularly useful to the instructor of a laptop-enhanced class. The students of a laptop-enhanced CS4 class can download the tablet PC algorithm corresponding to the data structure of the day and can use the algorithm in a structured class activity designed by the instructor. Students can be actively learning about *heapsort*, for example, rather than passively listening to the instructor lecture on it.

After the semester ends, the *best* student will be invited, possibly for as a senior thesis project, to collect all of the projects and organize all source code and documentation into a single website. Part of this task will also be to complete any of the projects that were either unsuccessful or not attempted. The course

website will be a future reference for anyone (CS1, CS2, CS4 and CPSC 481/681 instructors and students) who is interested in working with the tablet PC programs. The URL will be provided to Microsoft to include in any database or webpage the company chooses.

In future semesters, tablet PC software will be developed for other courses, such as *Computer Organization* or *Computer Science Theory*. For a theory course, as an example, we envision developing software that will allow a student to build finite state automata and demonstrate the activity generated within triggered by input of strings from a predefined alphabet provided by the student. For a computer organization course, students may draw a register set, a CPU, and a memory, interconnect them using one or more buses, and observe what the register transfers occur when different assembly-language instructions are executed.

Schedule of Tasks

The following schedule describes the tasks to be accomplished as part of this project:

- Design course, exercises and project descriptions (summer 2005)
- Teach CPSC 481/681 first time, collect formative and summative assessments (fall 2005)
- Collect all tablet PC algorithms, organize course website, write paper(s) for computer education conference(s) (spring/summer 2006)

This project will conclude at the end of summer 2006 and all materials produced, including research papers, source code and documentation will be made available to Microsoft Corporation and all interested educators. Questions, suggestions and comments are welcome.

.

Figure 1: Inorder traversal of a binary tree.