NSF
CCLI Project Showcase

SIGCSE 2006
Houston, Texas

March 1-5, 2006

Booths W & Y
Ballroom of the Americas

National Science Foundation
Course, Curriculum and Laboratory Improvement Program

Program at a Glance

Thursday, 10:30 a.m.—12:00 p.m.

Program Visualization using Virtual Worlds (page 4)
Stephen Cooper, St. Joseph's University
Wanda Dann, Ithaca College
Barbara Moskal, Colorado School of Mines
Randy Pausch, Carnegie Mellon University

Online Programming Tutors for Computer Science I (page 5)
Amruth Kumar, Ramapo College of New Jersey

Extending the Next Generation Robot Laboratory to Increase Diversity in Undergraduate CS Programs (page 6)
Maria Gini, University of Minnesota
Karen Sutherland, Augsburg College
Janice Pearce, Berea College

Friday 10:30 a.m.—12:00 p.m.

The UC-WISE project: Online Curricula for Monitored, Closed-lab First-year CS Courses (page 7)
Mike Clancy, University of California at Berkeley
Marcia Linn, University of California at Berkeley
Nate Titterton, University of California at Berkeley

Integrating Algorithm Visualization into Computer Science Education (page 8)
Scott Grissom, Grand Valley State Univ
Myles McNally, Alma College
Thomas Naps, Univ of Wisconsin, Oshkosh

CS 0.5: A New Approach to Introductory Computer Science for Majors (page 9)
Robert H. Sloan, University of Illinois at Chicago
Patrick Troy, University of Illinois at Chicago

(this session continued on the next page)

Program at a Glance

Friday 10:30 a.m.—12:00 p.m. (continued)

Intelligent Tutoring System for CS-I and II Laboratory (page 10)
Jungsoon Yoo, Middle Tennessee State University
Sung Yoo, Middle Tennessee State University
Chrisila Pettey, Middle Tennessee State University
Judy Hankins, Middle Tennessee State University
Cen Li, Middle Tennessee State University
Suk Seo, Middle Tennessee State University

Friday 2:00 p.m.—3:30 p.m.

jGRASP: A Framework for Integrating Visualizations of Software (page 11)
James Cross, Auburn University
Dean Hendrix, Auburn University
David Umphress, Auburn University

Project MLExAI: Machine Learning Experiences in AI (page 12)
Ingrid Russell, University of Hartford
Zdravko Markov, Central Connecticut State University
Todd Neller, Gettysburg College

A Radical Approach to Teaching Object-oriented Programming (page 13)
Kathryn E. Sanders, Rhode Island College
Ann Moskol, Rhode Island College

Increasing Interaction and Visualization in the Computability Course (page 14)
Rakesh M. Verma, University of Houston
Pavan Podila, University of Houston
Saquib Hakim, University of Houston

Saturday 10:30 a.m.—12:00 p.m.

Due to logistical reasons, the Saturday session has been cancelled.

Program Visualization using Virtual Worlds

Thursday, 10:30 a.m.—12:00 p.m.

Stephen Cooper, St. Joseph's University
Wanda Dann, Ithaca College
Barbara Moskal, Colorado School of Mines
Randy Pausch, Carnegie Mellon University

This project uses a high-impact, high-interest program visualization environment, Alice, to introduce a strong core of fundamental programming concepts (objects, classes, methods, decisions, variables, parameters, loops, etc.) and problem-solving design techniques to beginning programmers. Results of our previous NSF proof-of-concept project provide evidence that this approach can significantly improve achievement and retention of at-risk CS-majors during their first year. We have anecdotal evidence that this approach captures student interest and motivates positive attitudes for women and minorities.

Our proof-of-concept grant yielded a prototype of instructional materials and a textbook that work synergistically with the software environment. The prototype materials have been pilot-tested and peer-reviewed. An objective of this project is to revise and expand the prototype material based on what we learned in the pilot. Other faculty members have successfully used our materials, but no formal testing beyond our pilot project has been done. A second objective is to use and formally test the revised materials at four regional test sites the first year and at least six additional sites in the second year. The regional test sites are different types of institutions (two major universities, an HBCU, and a liberal arts college) serving students with diverse backgrounds and career goals. We will provide professional development for a small core of regional leaders at four test sites during the first year and then expand the scope to additional faculty at schools in the nearby areas. It is anticipated that this will create a community of faculty teaching with Alice.

Online Programming Tutors for Computer Science I

Thursday, 10:30 a.m.—12:00 p.m.

Amruth Kumar, Ramapo College of New Jersey

Problets are online tutors for C/C++/Java/C# programming. Each problet 1) presents problems; 2) grades the student's answer; 3) provides instant feedback; and 4) records the student's performance.

Problems: No two problems presented by a problet are alike. Problets can present problems ad-infinitum. They adapt problems to the learning needs of the student, so students solve problems on only the concepts they do not understand.

Feedback: Problets not only tell the student whether his/her answer is correct or wrong, but also explain the step-by-step execution of the problem code. This latter feature helps students learn from their mistakes.

Topics: Currently, problets are available for: arithmetic expressions, relational expressions, variables and scope, if and if-else statements, while loops, for loops, functions, and C++ pointers. Problets present problems on debugging programs, predicting the output of programs and evaluating expressions.

Use: Problets are designed to be used as supplements to classroom instruction and complements to programming projects assigned in Computer Science I course. They can be used for tutoring (in-class or self-paced), or testing.

Pedagogy: Problets help students learn programming concepts by solving problems. They promote active learning.

For the Instructor: Instructors who assign problets in their class can find out how well each student did, how well the class did, and the concepts that are not yet clear to each student/class.

Logistics: Problets are free for educational use. They run on any recent Java-enabled browser. To find out more, please visit: http://phobos.ramapo.edu/~amruth/grants/problettes/home.html. If you are interested in using the problets, please contact: Amruth Kumar, amruth@ramapo.edu

Extending the Next Generation Robot Laboratory to Increase Diversity in Undergraduate CS Programs

Thursday, 10:30 a.m.—12:00 p.m.

Maria Gini, University of Minnesota
Karen Sutherland, Augsburg College
Janice Pearce, Berea College

By developing new CS course materials starting at the very beginning of the curriculum, this collaborative project is designed to increase the interest of women and other underrepresented groups in studying Computer Science and to provide them with tools that will help throughout their undergraduate years. The course materials are based upon Pyro, an easy to use programming system for robots written in Python and are designed for multiple robotics platforms such as Sony AIBOs and eROSIs. eROSIs are an inexpensive but powerful miniature robotic platform developed at the University of Minnesota and which are being considered for the monitoring of highly sensitive areas of populations such as elderly or disabled patients in residential care.

The project team comes from different types of institutions, so the materials created are being tested in different situations and should prove readily adaptable for a wide variety of educational environments.

The UC-WISE project: Online Curricula for Monitored, Closed-lab First-year CS Courses

Friday, 10:30 a.m.—12:00 p.m.

Mike Clancy, University of California at Berkeley
Marcia Linn, University of California at Berkeley
Nate Titterton, University of California at Berkeley

ꕥ • ꕥ

The UC-WISE project (University of California Web-based Instruction for Science and Engineering) aims

- To provide technology and curricula for laboratory-based higher-education courses that incorporate online facilities for collaboration, inquiry learning, and assessment;
- To allow instructors to customize courses, prototype new course elements, and collect review comments from experienced course developers.

A UC-WISE course trades lecture time for online supervised lab time. Activities in a UC-WISE curriculum include online discussions, programming exercises, reading of Web-delivered text, reflection notes, journal entries, quizzes, and "gated collaborations" where students critique their peers' responses to a seed topic. Lab instructors may view some student work in real time, enabling immediate targeted tutoring where appropriate.

We have so far produced lab-based UC-WISE curricula for three courses. Our Scheme-based introductory programming course has been run in this format since spring 2003. We have piloted a lab-based curriculum for a new introductory programming course for engineering majors, and have run two offerings of our lab-based CS 2. NSF is supporting collaboration with colleagues at three other U.C. campuses to develop curricula for Java-based introductory courses.

Our research explores several aspects of student learning:

- Effects of UC-WISE activities not in traditional-format courses, and of various kinds of immediate feedback for students;
- Student misconceptions revealed by the rich variety of activities in UC-WISE courses;
- Relative benefits of online vs. face-to-face collaborative activities;
- Exploration of tantalizing evidence of differential benefits of UC-WISE courses for males vs. females.

Integrating Algorithm Visualization into Computer Science Education

Friday, 10:30 a.m.—12:00 p.m.

Scott Grissom, Grand Valley State University
Myles McNally, Alma College
Thomas Naps, University of Wisconsin, Oshkosh

Despite the abundance of algorithm visualization (AV) tools now available, their promise as a pedagogical tool remains largely unfulfilled. Our goal is to increase the impact of AV on CS education by addressing previous barriers. We are developing a collection of online visualizations that support active learning outside of the classroom supported by a student laboratory manual (http://jhave.org)

Algorithm visualizations (AV) use computer graphics to depict the actions of an algorithm. The reasons for their popularity are threefold. For researchers in the field of AV, there is the challenge of developing new visualization techniques. For practitioners, AV can help in the process of designing and debugging algorithms. For computer science students, AV holds promise to help them understand algorithms more easily and in greater depth. Despite their potential, their promise as a pedagogical tool is largely unfulfilled.

Our primary goal is to increase the impact of AV on CS education by addressing previous barriers. We will emphasize pedagogy by merging existing AV resources with proven learning techniques. Many of the techniques have been applied in limited contexts. We have the following specific goals:

- Identify and evaluate effective strategies that use AV to encourage active learning outside of class.
- Develop Web-based AV resources to support learning of data structures and algorithms in a variety of CS courses. A platform-independent Web solution will make it easier for faculty to review and adopt (http://jhave.org)
- Publish a student laboratory manual that supports the online visualizations with pre-lab and post-lab activities.
- Provide summer training workshops to help faculty adopt our materials and develop their own. The next workshop will be June 13-16, 2006 in Grand Rapids, MI.

CS 0.5: A New Approach to Introductory Computer Science for Majors

Friday, 10:30 a.m.—12:00 p.m.

Robert H. Sloan, University of Illinois at Chicago
Patrick Troy, University of Illinois at Chicago

There are often problems when students enter a course with widely different experience levels with key course topics. If the material is covered too slowly, those with greater experience get bored and lose interest. If the material is covered too quickly, those with less experience get lost and feel incompetent. We have found this to be the case for incoming students into our Computer Science Major. This situation has led to the creation of CS 0.5, an introductory Computer Science course to target those CS majors who have little or no background with programming. Our goal is to provide these students with an engaging curriculum and prepare them to keep pace in future courses with those students who enter with a stronger background.

Following the lead of Mark Guzdial's work on using media computation for non-majors at Georgia Tech, we are using media computation as the tools to provide this engaging curriculum. We provide a preliminary report on our success in using a media computation course for CS 0.5.

Intelligent Tutoring System for CS-I and II Laboratory

Friday, 10:30 a.m.—12:00 p.m.

Jungsoon Yoo, Middle Tennessee State University
Sung Yoo, Middle Tennessee State University
Chrisila Pettey, Middle Tennessee State University
Judy Hankins, Middle Tennessee State University
Cen Li, Middle Tennessee State University
Suk Seo, Middle Tennessee State University

A Web-based adaptive tutoring system which dynamically adapts to each student's needs and gives a student immediate feedback is being developed for our CS-I and CS-II closed laboratories. The system currently contains the question tutor, the program tutor, and the course management components. The tutoring components help students learn programming concepts through hands-on, self-paced exercises. The course management component helps teachers prepare and maintain the lab materials. Experiments have been conducted to evaluate the effectiveness of this new tutoring system and promising preliminary results were obtained.

jGRASP: A Framework for Integrating Visualizations of Software

Friday, 2:00 p.m.—3:30 p.m.

James Cross, Auburn University
Dean Hendrix, Auburn University
David Umphress, Auburn University

Software visualizations are recognized mechanisms for teaching fundamental programming concepts, yet their classroom potential is seldom fully realized. Lack of integration of a variety of visualizations into a common, easily used framework is a primary obstacle. This project proposes to remove this barrier by providing a robust framework for integrating visualizations such as control structure diagrams, UML class diagrams, and dynamic data structure views. The fully developed framework will be based on jGRASP, a lightweight prototype environment which has been extremely well received by students, educators, and publishers.

Approach. Static and dynamic visualizations will be generated directly from the student's program before, during, and after its execution. Baseline object viewers for Java collections classes will be provided, and a view builder and API will facilitate Plug-in object viewers to provide additional visualizations for specialized data structures. The framework uniquely integrates these object viewers with the debugger to produce "live" visualizations of the student's program.

Impact. We expect this project to have a significant impact on computing education. The result will be an important new tool that, while providing an integrated development environment, also exploits the power of visualizations to enhance computer science and software engineering education. jGRASP will be freely available to the computing research and education community. Previous versions of jGRASP have been downloaded by over 100 institutions, and we expect the full-scale jGRASP resulting from this project to be downloaded and used by these as well as other institutions around the world.

Project MLExAI: Machine Learning Experiences in AI

Friday, 2:00 p.m.—3:30 p.m.

Ingrid Russell, University of Hartford
Zdravko Markov, Central Connecticut State University
Todd Neller, Gettysburg College

It is generally recognized that an undergraduate introductory Artificial Intelligence (AI) course is challenging to teach. This is, in part, due to the diverse and seemingly disconnected core topics that are typically covered. The paper presents work on a collaborative project funded by the National Science Foundation to address this problem and to enhance the student learning experience in the course. Our work involves the development of an adaptable framework for the presentation of core AI topics through a unifying theme of machine learning. A suite of hands-on term-long projects are developed, each involving the design and implementation of a learning system that enhances a commonly-deployed application. The goals of project MLExAI are to (1) enhance the student learning experience in the AI course, (2) increase student interest and motivation to learn AI by providing a framework for the presentation of the major AI topics that emphasizes the strong connection between AI and computer science and engineering, and (3) highlight the bridge that machine learning provides between AI technology and modern software engineering. We will present an overview of our model and the projects being developed. Our preliminary experiences incorporating these modules into our introductory AI courses will also be presented.

A Radical Approach to Teaching Object-oriented Programming

Friday, 2:00 p.m.—3:30 p.m.

Kathryn E. Sanders, Rhode Island College
Ann Moskol, Rhode Island College

ꕥ • ꕥ

We are adapting an introductory programming course originally developed at Brown University for use at Rhode Island College. The course is characterized by extensive use of graphics, very early object-orientation (encapsulation, inheritance, and polymorphism are all taught before such traditional topics as arithmetic and flow of
control), and a strong emphasis on design.

The original materials, thoroughly tested during the preceding 5-6 years at Brown, included several chapters of a text, a complete set of slides, and programming assignments. We are adding weekly in-class labs, groupworks, and homeworks (and in addition, Dr. Sanders and Professor van Dam of Brown are co-authors of a recently published complete text based on this approach).

We are also developing an approach to assessing our materials that could be applied to any introductory programming course.

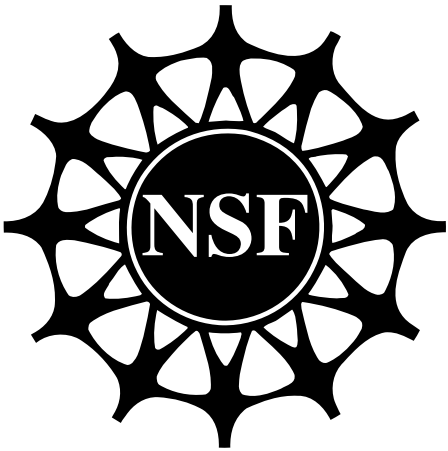
Increasing Interaction and Visualization in the Computability Course

Friday, 2:00 p.m.—3:30 p.m.

Rakesh M. Verma, University of Houston
Pavan Podila, University of Houston
Saquib Hakim, University of Houston

ꕥ • ꕥ

Teaching Theory of Computation and learning it are both challenging tasks. Moreover, students are not sufficiently interested/motivated to learn this material since: (i) they believe that the material is dated and of little use and (ii) it is too abstract and difficult. To counter the first perception, we have developed materials to illustrate the breadth of finite automata concepts. To overcome the second problem we have: enhanced and integrated visualization software and historical background into newly-developed materials including homeworks and slides for lectures. Most of the materials are available at the web site (http://www.cs.uh.edu/~rmverma/3340new/html/index.html) for the course that we developed. Our overall experience has been positive. Remarkably, since the course was revamped several students were motivated to pursue projects in Automata theory, including assisting on the web site and the course materials, and many of them went on to graduate school.



The NSF CCLI showcase is a yearly event at the SIGCSE conference. If you are working on an NSF CCLI grant, or have recently completed one and would like to present at the showcase, please contact Aaron Bloomfield at asb (at) cs (dot) virginia (dot) edu. The selection process will begin in the early fall.