

# SIGCSE 2006 Workshops

The following workshops were available to attendees of the SIGCSE 2006 Symposium for a nominal fee.

## Wednesday Workshops, 7:00 p.m. to 10:00 p.m.

### 1. Abstraction — is it teachable?

Jeff Kramer, *Imperial College London*  
Orit Hazzan, *Technion – Israel Institute of Technology*

Based on the acknowledgment that abstraction is recognized as a key concept in computing, the workshop examines the role of abstraction in computer science and software engineering in general and addresses the teaching of abstraction in particular. Specifically, the following topics will be discussed:

- The undergraduate computer science and software engineering education needs for abstract thinking;
- Pedagogical guidelines of teaching abstract thinking to computer science and software engineering undergraduate students;
- Specific activities for teaching abstract thinking in undergraduate computer science and software engineering programs.

### 2. Integrating Privacy Content Into a Database Course

Florence Appel, *Saint Xavier University*

This workshop takes an ethics across the curriculum approach to integrating privacy issues into introductory database courses. Participants will work through a set of *privacy modules* (lecture notes, in-class and homework exercises, test questions) intended for systematic implementation throughout the database design thread of a course. The modules are written to address relevant privacy issues at each stage of the database design process. Participants will receive brief instruction on fundamental applied ethics concepts, and pointers for the seamless integration of privacy content. For the major part of the workshop, they will work in groups to simulate classroom activities.

### 3. Enhancing Learning in CS1 with Roles of Variables

Jorma Sajaniemi, *University of Joensuu*  
Pauli Byckling, *University of Joensuu*

Roles of variables describe stereotypic uses of variables; only eleven roles are needed to cover practically all variables in novice-level programming. Roles provide teachers a new pedagogical tool and students a new conceptual framework that results in better programming skills. This workshop introduces the role concept and individual roles applicable within object-oriented, procedural, and functional programming, and describes how roles can be utilized in teaching. After the workshop, participants will be able to teach roles and develop basic role-based programming strategies in students. The workshop is intended for teachers of elementary programming courses.

### 4. Object-Oriented Design Festival

Dung Nguyen, *Rice University*  
Stephen Wong, *Rice University*  
Eric Cheng, *Rice University*  
Mathias Ricken, *Rice University*

Object-oriented (OO) programming begins with analysis and design that produce a model describing the objects in the problem domain, their relationships, creation and interactions. The workshop covers fundamentals of OO analysis and design such as abstraction, separation of variants from invariants and decoupling of system components, via appropriate applications of composition, inheritance, polymorphism, and design patterns. The workshop will progress from a small design example illustrating the principles to a larger design problem to be solved by small teams of participants. Their solutions will be discussed in terms of design goals and compared against a solution provided by the presenters.

## 5. **Computer Security Essentials (Part 1): System Footprinting and Vulnerability Assessment**

Paul J. Wagner, *University of Wisconsin – Eau Claire*  
Daren Bauer, *University of Wisconsin – Eau Claire*  
Thomas Paine, *University of Wisconsin – Eau Claire*  
Andrew Phillips, *University of Wisconsin – Eau Claire*  
Jason Wudi, *University of Wisconsin – Eau Claire*

This is the first of two hands-on workshops for CS educators seeking to develop curricula in computer security. We provide guided hands-on instruction on various Windows and Linux based tools commonly used for gathering information about, and assessing the vulnerability of, other systems. Participants will experiment with these tools as the presenters guide them through typical tool use scenarios. The session concludes with an information gathering exercise on an isolated network. Participants will use laptops running both Windows and Linux images pre-configured with common security "holes" so that they may experience first-hand the process of information gathering and vulnerability detection.

## 6. **Women in Computing Education Oral History Project : Oral History Interview Training**

Barbara Boucher Owens, *Southwestern University*  
Vicki L. Almstrum, *University of Texas at Austin*  
Lecia J. Barker, *University of Colorado*

This workshop is designed to train interviewers for the Women Computing Educators Oral History Project. This workshop will provide background, techniques, guidelines, equipment selection criteria, and demonstrations and practice in conducting interviews. This endeavor should eventually result in a significant collection of narratives of women computing educators, a relatively inexpensive and enduring format that could provide passive mentoring and highlight women in the field as role models. The project will also be a mechanism for preserving the history of the female pioneers in computing education and a model for broader projects to capture the history of male computing educators.

## 7. **Active and Cooperative Learning Techniques for Computer Science Education**

Jeffrey J. McConnell, *Canisius College*

Active and cooperative learning provides a powerful mechanism to enhance depth of learning and increase material retention. Active and cooperative learning gets students involved with the material rather than passively listening to a lecture. This workshop will use introductory material on active and cooperative learning for a number of activities that will give participants direct experience with and the chance to observe these techniques in action. There will also be opportunities for open discussion of situations that participants may have already encountered and for the development of new activities for the classroom.

## 8. **Hands-on Activities for Learning Discrete Mathematics Concepts**

Peter B. Henderson, *Butler University*  
Susanna S. Epp, *DePaul University*  
William Marion, *Valparaiso University*

*"Tell me and I'll forget. Teach me and I'll remember. Involve me and I'll learn."* Hands-on activities, with physical objects or computer-based tools, engage students in the learning process. They help students understand relevant discrete mathematics concepts and connect topics in discrete mathematics and computer science. Participants will learn to use everyday objects, cards, coins, dice, paper, string, straws, etc. to explore counting, probability, recursion, graphs, and logic. Computer applets, covering a broad spectrum of discrete mathematics topics, and examples and activities using the functional language Scheme will also be presented.

## 9. **Using Eclipse to Develop and Debug Java Programs**

Barry Burd, *Drew University*

The Eclipse development environment is great for both large and small classroom projects. A recent survey says that over 50% of all professional Java developers use Eclipse, and the number is growing. Eclipse is open source software. It's free to use, and free to modify. This workshop begins with an overview for Eclipse "newbies." The workshop's focus then shifts to program development and debugging. Eclipse's refactoring tools help the student develop and refine an application's code. Then, with the aid of Eclipse's debugger, the student analyzes the code to detect and correct errors.

## 10. **Multimedia Projects for CS1 and CS2**

Mark Guzdial, *Georgia Institute of Technology*  
Barbara Ericson, *Georgia Institute of Technology*

Constructing and manipulating media is a motivating context for students in CS1 and CS2.. Modern computers are capable of interesting media effects in reasonable running time, using simple and obvious algorithms that fit within the constraints of introductory courses' curricula. In this workshop, we will present algorithms (mostly in Python) and working code (in Java) for the creation and manipulation of sound, image, MIDI music, and video data. Example techniques will include linked lists of MIDI music, sound splicing and reversing, chromakey ("blue/green screen") image effects, animation, and Photoshop-like image filters.

## 11. **jGRASP: An Integrated Development Environment with Visualizations for Teaching Java in CS1, CS2, and Beyond**

James H. Cross II, *Auburn University*  
Dean Hendrix, *Auburn University*  
David Umphress, *Auburn University*

jGRASP is an integrated development environment that provides automatic generation of visualizations to improve the comprehensibility of software. These visualizations, which are particularly well suited for CS1 and CS2 students learning Java, include Control Structure Diagrams, UML Class Diagrams, and dynamic Object Views (including arrays, ArrayList, LinkedList, HashMap, and TreeMap). The workshop will include tutorials with example programs to demonstrate how instructors can improve the learning and programming experience of their students with jGRASP. Instructors are encouraged to bring programs from their own courses and experiment with the visualizations of these during the workshop. jGRASP is freely available at [www.jgrasp.org](http://www.jgrasp.org).

### **Friday Workshops, 7:00 p.m. to 10:00 p.m.**

## 12. **Travels in "DNA Land" — Approaching DNA Sequence Analysis Through Word Play**

Mark D. LeBlanc, *Wheaton College*  
Betsey D. Dyer, *Wheaton College*

In a series of team-taught mini-lectures followed by hands-on activities, participants will learn how to introduce, motivate, and explain DNA sequence analysis by engaging in "word play" using regular expressions on English texts and then DNA sequence. Faculty wishing to experience an introductory but focused look at DNA sequence analysis and its place as a rich source of state-of-the-art applications are especially encouraged to attend. Participants will take away a series of classroom tested specifications used in a variety of computer science courses at various levels in the curriculum and consider strategies for starting collaborations with biology and other science faculty at their home institution. Participants will alter some Perl, but no prior knowledge of Perl is required. Participants will bring their own laptops pre-loaded with workshop materials.

## 13. **Kinesthetic Learning in the Classroom**

Andrew Begel, *UC Berkeley*  
Rebecca Bates, *Minnesota State Univ., Mankato*  
Steven A. Wolfman, *Univ. of British Columbia*

This workshop will focus on kinesthetic learning activities, i.e., physically engaging classroom exercises. These might, for example, teach recursion by simulating the Towers of Hanoi with students instead of disks, or demonstrate the efficacy of various hash functions by sorting students by (numerical) characteristics such as birth month. The workshop will begin with a brief kinesthetic learning activity to motivate the value of these activities. We will follow with a variety of examples, and discuss how to deploy these in a classroom. Most of the workshop will be devoted to facilitated group work to help the participants design, test and evaluate their own activities. All KLAs will be posted to <http://sequoia.cs.berkeley.edu/kla/>.

## 14. **Computer Security Essentials (Part 2): Intrusion Detection and System Defense**

Paul J. Wagner, *University of Wisconsin – Eau Claire*  
Daren Bauer, *University of Wisconsin – Eau Claire*  
Thomas Paine, *University of Wisconsin – Eau Claire*  
Andrew Phillips, *University of Wisconsin – Eau Claire*  
Jason Wudi, *University of Wisconsin – Eau Claire*

This is the second of two workshops for CS educators developing curricula in computer security. Participation in workshop part one is recommended as a prerequisite. We provide guided hands-on instruction and experimentation on both defensive techniques and the understanding of exploits for the purpose of better defending systems. The session concludes with a hands-on

exercise giving the participants an opportunity to participate in a carefully constructed and monitored cyberwar scenario; i.e. the participants will harden their systems, identify potential exploits and threats, and work to understand the mindset of the attacker by identifying weaknesses in all systems on the network.

### **15. Emerging Robotics Resources in Undergraduate CS**

Zach Dodds, *Harvey Mudd College*  
Doug Blank, *Bryn Mawr College*  
Paul Rybski, *Carnegie Mellon University*

This workshop will be a hands-on opportunity to assess several emerging software, hardware, and curricular resources that further the capabilities and roles of robots within the undergraduate CS curriculum. We will present and guide participants in trying (1) Pyro, a free, python-based software abstraction that facilitates interaction with robots and robot simulators for students of all experience levels; (2) Sony's AIBO and CMU's TeRK, medium- and low-cost hardware platforms that are seeing increased use at the undergraduate level; (3) several curricular modules and student-research projects based on these resources.

### **16. Introduction to Microsoft .NET for Academia**

Joe Hummel, *Lake Forest College*

Microsoft .NET is an exciting framework for programming not only on the Windows platform, but MacOS and Linux as well. This workshop will introduce attendees to .NET programming, in particular with regards to building console-based, GUI, and database-driven applications appropriate for students. Both command-line tools (free) and Visual Studio .NET will be demonstrated. The purpose of this workshop is to introduce .NET, allowing attendees to evaluate its use in a CS or IS curriculum. PowerPoint presentations will be example-based using VB and C#, and suitable for student distribution; attendees will have access to electronic copies of all workshop materials.

### **17. Regional Celebrations of Women in Computing – Best Practice**

Bettina Bair, *Ohio State University*  
Cinda Heeren, *University of Illinois*  
Peter B. Henderson, *Butler University*  
Cary Laxer, *Rose-Hulman Institute of Technology*  
Gloria Childress Townsend, *DePauw University*

Hosting a regional celebration can be a powerful way of creating connections among women and underrepresented groups in computing. Yours can vary in size, focus and length. Based on our experience, we can help you establish realistic goals, identify your target audience, create a feasible timeline and budget, secure financial support, select and manage your conference site, promote your event, solicit for participation, automate registration and assessment. You'll also learn what kinds of problems to expect and how to deal with them. You'll receive a 'Best Practices' booklet and worksheets for planning your own regional celebration.

### **18. Developing with Service-Oriented Architectures**

Brian W Loomis, *Microsoft Corporation*

This workshop will describe the tools, processes and techniques used in integrating web services and applications in a service-oriented architecture. Materials will be presented using business examples but will be appropriate for teaching an upper-level course in internet technologies, distributed applications, or systems architecture. We will talk specifically about the internet standards (XML, XSD, web services) and the architectural framework of service-oriented architecture (SOA) and demonstrate typical usage. We will discuss the emerging technologies around web service enhancements (WS-E, security, transactions) as well as software modeling (UML) techniques which provide basic functionality for integrating applications.

### **19. Software Development for the Tablet PC**

Jeffrey L. Popyack, *Drexel University*  
Bruce Char, *Drexel University*

Tablet PC's feature a stylus capable of inputting a data type called Ink. Ink has many characteristics that allow freehand pen annotation, editing, handwriting recognition, and text search. Microsoft's Tablet PC Software Development Kit (SDK) allows development of Windows-based applications for Tablet PC's with Visual Studio .NET, which includes a full set of graphical user interface and Ink manipulation tools. This introductory-level workshop includes a brief overview of Tablet PC development environment, resources available for developers, and instruction on using the SDK, with suggestions for usage in software design and team projects throughout the computer science curriculum.

## 20. Teaching Students to do Effective Unit Testing

Barry Burd, *Drew University*

Testing is an essential part of the software development process. So we should teach students a solid testing methodology. One such methodology is *unit testing*. With unit testing, the programmer tests each piece of code before adding the code to an application. We should also give students tools to automate the testing process. JUnit is the most widely used unit testing tool among professional Java programmers. Using JUnit, a programmer creates rigorous tests, instruments existing code, and runs a formal test suite. JUnit itself is free, open source software. This workshop covers JUnit and its use in undergraduate computer science courses.

## 21. Learning to program with Alice

Stephen Cooper, *Saint Joseph's University*  
Wanda Dann, *Ithaca College*

This introductory workshop, designed for instructors with minimal to no prior experience with Alice, offers hands-on experience programming with Alice. Alice is a powerful program visualization tool enabling students to “see” objects and work with object-oriented programming. Participants will learn how to use Alice to build virtual worlds and how to use this approach in introductory-level computing courses (introductory programming for majors, programming for non-majors, computer literacy, etc.) at the college or high-school level. Participants will receive a CD containing the latest version of the software and sample virtual worlds developed as part of NSF-0126833/NSF-0339734.

## 22. Using LEGO Robots for Traditional Projects in AI

Amruth N Kumar, *Ramapo College of New Jersey*

In this workshop, we will discuss how to use LEGO robots to assign projects on traditional topics in AI. We will discuss robot projects on topics such as search, expert systems and neural networks: the design, logistics, implementation options, and guidelines for hardware, software and props for the projects. Participants will work through the Java solutions of two projects - complete, compile, download and execute them on LEGO robots to get a hands-on feel. This workshop will be of interest to instructors who would like to use robots for projects in their AI course. Knowledge of Java is beneficial.

## 23. Integrating Algorithm Visualization into Computer Science Courses

Tom Naps, *U Wisconsin. Oshkosh*  
Myles McNally, *Alma College*  
Scott Grissom, *Grand Valley State University*

Participants will learn how to use materials developed by the presenters under a NSF EMD grant. These materials, along with the supporting software system (JHAVÉ), emphasize pedagogy instead of technology. Topics covered will include best practices in using algorithm visualization (AV), fostering student engagement with AV, and the design of effective instructional AV. Hands-on activities will allow exploration of numerous visualizations already developed and, to a lesser degree, creating new visualizations in the JHAVÉ environment. Participants will receive the draft of an AV-based lab manual and be invited to a more intensive follow-up summer workshop funded by the NSF grant.

## 24. Online Programming Assignments with WeBWorK1

Olly Gotel, *Pace University*  
Richard Kline, *Pace University*  
Christelle Scharff, *Pace University*  
Andrew Wildenberg, *Cornell College*

WeBWorK (<http://webwork.math.rochester.edu>) is a free open-source formative web-based assessment system to generate, deliver and grade homeworks, and distribute their solutions. It emphasizes online feedback and retesting opportunities to master learning. This workshop will explore the use of WeBWorK for teaching programming fundamentals. Attendees will experiment with the student and instructor WeBWorK interfaces, and write their own customized WeBWorK problem sets (with their solutions, grading schemas and deadlines) in the Problem Generating (PG) macro language. The developed WeBWorK problems will go from simple True/False, multiple-choice and matching problems, to more sophisticated problems testing program correctness using a WeBWorK plug-in interfacing WeBWorK with JUnit.

## Saturday Workshops, 3:30 p.m. to 6:30 p.m.

### 25. Secure Programming: Buffer Overflows and Other Common Mistakes

James Walden, *Northern Kentucky University*  
Charles E. Frank, *Northern Kentucky University*  
Laurie Werner, *Miami University Hamilton*

Despite the discovery of thousands of software vulnerabilities each year, few computer science programs teach secure programming techniques. This workshop introduces participants to common programming mistakes that produce security vulnerabilities, including buffer overflows, integer overflows, format string problems, directory traversal vulnerabilities, and SQL injection flaws. Examples of vulnerabilities and solutions will be shown in C, C++, and Java. We will also demonstrate how to find flaws using the Fortify Source Code Analysis Suite and discuss the use of secure design principles to create software that remains secure despite the presence of implementation flaws.

### 26. Teaching Mobile and Ad-hoc Networking using Simulation

Chris McDonald, *The University of Western Australia*

This workshop will demonstrate that students' understanding of mobile and ad-hoc wireless networking can best be developed and assessed through quality, interactive, simulations. Classroom-tested material will demonstrate detection and recovery from data corruption and loss, collision detection and avoidance, table-driven and on-demand routing algorithms, protocols for battery power minimization, and the security of wireless networks. The workshop draws on our many years' teaching experience with the simulation of wide-area, local-area, and mobile and ad-hoc wireless environments, in both large undergraduate courses and graduate projects. Faculty will be introduced to exercises and assessments suitable for undergraduate open- and closed-laboratory sessions, and capstone projects.

### 27. Adding Software Testing to Programming Assignments

Stephen H. Edwards, *Virginia Tech*

This workshop provides a practical introduction to how one can incorporate software testing activities as a regular part of programming assignments, supported by live demonstrations. It presents five different models for how one can incorporate testing into assignments, provides examples of each technique, and discusses the corresponding advantages and disadvantages. Approaches to assessment—using testing to assess student code, assessing tests that students write, and automated grading—are all discussed. Advice for writing “testable” assignments is given. Participant discussions are encouraged.

### 28. Designing Assignments and Exams using Bloom's Taxonomy

Raymond Lister, *University of Technology, Sydney*

Bloom's taxonomy provides a useful framework for designing course objectives, and is especially useful for designing assignments and exams. Qualitatively different grading tasks are set for each passing grade, with the differences in the nature of the tasks reflecting the differing levels of Bloom's taxonomy. After the workshop presenter has presented his own experience with using the approach on two different courses – one CS1, the other an introduction to databases – the workshop will move to an interactive mode where attendees will sketch a Bloom-based approach for their own course, and discuss it with the presenter and other workshop attendees.

### 29. Java: an Eventful Approach

Kim B. Bruce, *Pomona College*  
Andrea Danyluk, *Williams College*  
Tom Murtagh, *Williams College*

In this workshop we describe an objects-first approach to teaching Java that introduces event-driven programming in the very first programming examples, introduces concurrent threads early, and uses graphics and animation extensively. We show how these seemingly advanced topics can be presented so that they are easy for introductory course students to grasp. We also show how our approach exposes students to object-oriented programming techniques more thoroughly than is possible in more traditional approaches. Our approach is supported by materials developed with NSF funding including extensive course notes, laboratory exercises, the objectdraw library, and a textbook published by Prentice Hall in the summer of 2005.

### **30. A Hands-On Approach to Formal Languages and Automata with JFLAP**

Susan Rodger, *Duke University*  
Thomas Finley, *Cornell University*  
Peter Linz, *University of California, Davis*

JFLAP is a package of tools for experimenting with automata theory and formal languages: automata (FA, PDA, Turing machines), grammars, regular expressions, L-systems and parsing methods (brute-force, LL, and SLR). In addition one can study conversions of one form of a language to another form (examples: DFA to regular expression, CFG to PDA). In this hands-on workshop, participants will get an overview of JFLAP and learn how to integrate it into an automata theory or compiler course. Participants will get hands-on experience solving problems with JFLAP. Bring a laptop and download JFLAP for free from [www.jflap.org](http://www.jflap.org).

### **31. Learning Agile Python**

Ariel Ortiz, *Tecnológico de Monterrey*

Python is an interpreted object oriented language with important dynamic features that make it a suitable tool for agile software development. Python is easy to learn. It encourages readability, modularity and code reuse. Python is available for all major platforms, and it's free! In this workshop, participants will be introduced to Python's main features and at the same time will learn and apply these agile development practices: test driven development, refactoring, and pair programming. Additionally, possible uses for Python in CS courses (CS1/2, Databases, Compiler Construction, Web Development, etc.) will be discussed. Attendees should be familiar with object oriented programming.

### **32. Digital Media: Why Do We Need It, and How Do We Teach It As REAL Computer Science?**

Jennifer Burg, *Wake Forest University*

This workshop will provide the participants with curriculum material that treats digital media as science, dividing the area into digital imaging, audio, video, and multimedia programming. The participants will be given extensive material including text-based chapters, interactive web-based demos, programming exercises, worksheets, and MATLAB exercises. Participants will discuss how to use this material in courses that teach digital media as computer science, covering topics such as sampling, quantization, aliasing, color models, compression, dynamic range, transforms, filters, and digital data communication. Capturing student interest in computer science via digital media and relating the science to hands-on assignments will also be discussed.

### **33. Design of Class Hierarchies: An Introduction to OO Program Design**

Viera K. Proulx, *Northeastern University*  
Kathryn E. Gray, *University of Utah*

This workshop presents a curriculum and pedagogy for designing of object-oriented programs that is driven by the structure of the class hierarchies that represent the data. Design recipes guide the programmer through the design process, provide a framework for pedagogical intervention, and support the programmer with measurable outcomes at each step. Testing, documentation, and class diagrams are integral components of this design process. Class hierarchies with composition, inheritance and self reference with examples of data precede any exposition to the design of methods. Dynamic dispatch and polymorphism arise naturally from the structure of data. Repeated code patterns motivate the design of abstractions through interfaces, function objects, and abstractions over data types.

### **34. Techniques and Strategies for Teaching with Alice**

Wanda Dann, *Ithaca College*  
Stephen Cooper, *Saint Joseph's University*

This workshop is targeted to faculty who have taught a course using Alice or are experienced in using Alice. Participants will learn how to use advanced features of Alice and strategies for supporting open-ended student projects. Example worlds will demonstrate the use of lists and arrays as well as techniques for scorekeeping and games. Each participant will receive a CD containing the latest version of the software and sample materials, developed as part of NSF-0126833 and NSF-0339734.

### **35. Using LEGO MindStorms in CS 1 and CS 2**

Myles McNally, *Alma College*  
Frank Klassner, *Villanova University*

This workshop focuses on using LEGO MindStorms as an active-learning platform in CS 1 and 2. After an introduction to the MindStorms platform, the workshop will focus on a hands-on introduction to tested series of laboratories for use in the first year of CS instruction. The laboratories range from basic programming concepts such as variables, computation and methods to

object-oriented concepts such as classes, interfaces, inheritance and polymorphism, and on to data structures such as stacks, queues and lists. Although Java-based, these projects can easily be adapted to C++. Participants will have priority to attend a follow-up NSF sponsored workshop the following summer.

**36. Testing and Understanding Student Programming Projects with the Marmoset Submission and Snapshot System**

William Pugh, *University of Maryland*

Jaime Spacco, *University of Maryland*

David Hovemeyer, *Vassar College*

Teaching programming courses is challenging because students make mistakes that are difficult to predict. You've probably seen student code and thought "I never dreamed someone would make that mistake!" What if you could find out what mistakes students are making as they work? Using the Marmoset submission and testing system, you can! In addition to providing constructive feedback to students and instructors, Marmoset collects detailed records of student code evolution, yielding a rich and largely unexploited source of data for software engineering research. The workshop will present Marmoset as practical classroom technology and a vehicle for research.

**37. Moodle - Getting Started with a Hands-on Tutorial in the Leading Open-Source LMS**

D.I. von Briesen, *Central Piedmont Community College*

This workshop will introduce the world's leading LMS and walk participants thru a hands-on demonstration of some of moodle's powerful features, including forums, quizzes, wikis, blogs, profiles, its block format, and the ease with which instructors can configure and customize moodle's pedagogically rich environment. Each participant will have access to their own course, which they can continue to develop after the workshop. Participants will also learn about the incredible community building around this open-source platform, with over 5000 installations in 120 countries. If you use the web, are interested in online learning, and care about education – come Moodle with us!

# SIGCSE 2006 Birds of a Feather

The following Birds-of-a-Feather sessions were organized for attendees of the SIGCSE 2005 symposium.

## **A Town Meeting: SIGCSE Committee on Expanding the Women-in-Computing Community**

Gloria Townsend, *DePauw University*

Paula Gabbert, *Furman University*

At SIGCSE 2004 we launched a SIGCSE committee to identify “best practices” for increasing the population of high school girls and college women enrolled in computing classes. This BOF will create an opportunity for those interested in this topic to come together and will help provide us with information we can incorporate in an upcoming report we will produce at the conclusion of our two-year charter.

## **Assessment: Tricks of the Trade**

Briana Morrison, *Southern Polytechnic State University*

What methods, tools, and processes are faculty using to do assessment of coursework and programs for accreditation purposes? What experiences have faculty had in their assessment efforts? This BOF will allow attendees to share their best practices in the area of assessment.

## **Secure Software Engineering**

Charles E. Frank and James Walden, *Northern Kentucky University*

Rose Shumba, *Indiana University of Pennsylvania*

Software engineers who understand secure software engineering practices are crucial to stem the flood of software vulnerabilities, yet few software engineering courses cover secure development techniques. This session will focus on secure development practices from requirements through testing and how to incorporate them into existing software engineering or information security curricula.

## **What is Research? Does work with undergraduates count? What about the scholarship of teaching?**

Joan Krone, *Denison University*

Joseph Hollingsworth, *Indiana University Southeast*

Stuart Zweben, *The Ohio State University*

The question “What is research?” is of great concern to junior faculty. There are a few institutions which have quantitative answers but the vast majority of colleges do not. It is not just junior faculty who struggle with this question. Certainly members of evaluating committees must pass judgment on the work of colleagues; mentors and chairs must discuss this question with their junior faculty; and all faculty must be concerned about this question throughout their careers.

## **Towards CS concept inventories: Assessing learning in Computer Science**

Craig Zilles, *University of Illinois at Urbana-Champaign*

One of the key challenges of doing research in education is the ability to assess student learning, so that the effectiveness of different pedagogical approaches can be compared. In Physics, an assessment tool called the Force Concept Inventory (FCI) has been shown to be effective at assessing whether students become “Newtonian thinkers” and, furthermore, that most traditional instructional approaches generally fail to convert students to “Newtonian thinkers”, even among those who score well on traditional exams. I will present the characteristics of the FCI that made it successful, and we will discuss whether and how similar instruments could be developed for computer science.

### **Using Alice to Attract New CS Majors**

James Caristi, *Valparaiso University*

Stephen Cooper, *Saint Joseph's University*

Wanda Dann, *Ithaca College*

This session will focus on the use of Alice in a course like CS0 for the purpose of attracting additional CS majors. We are especially interested in brainstorming ideas for using Alice to attract women into CS. Participants can share their experiences, present Alice worlds that would be helpful, raise concerns or difficulties, or otherwise participate in the discussion. Ideas generated will be collated and posted on the <http://www.aliceprogramming.net/> website.

### **Undergraduate Computational Science/Computational Biology Education**

Angela Shiflet, *Wofford College*

Computational Science is a fast-growing field at the intersection of the sciences, computer science, and mathematics and includes such areas as computational biology and bioinformatics. Computer science departments are in excellent positions to be instrumental in the development of undergraduate computational science. Moreover, addition of such application areas can be beneficial to computer science majors and can attract students, especially women, to computer science classes. Discussions in this BOF will focus on formation of such programs, curricula, internships, resources, experiences and challenges

### **Innovative Approaches to Broadening Computer Science**

Owen Astrachan and Jeffrey Forbes, *Duke University*

Computer science courses fail to attract a wide constituency of students, especially those intending to major in other disciplines such as biology, economics, public policy, mathematics, and other natural and social sciences. Despite the wide applicability and impact of computing on these fields, a common perception exists that computer science is an exercise in programming without broad applications and without an intellectual foundation that matters to these disciplines. This BOF will discuss alternative approaches to introducing computer science to students who we think don't know they might be interested in computer science.

### **Are Students Learning What We Are Teaching?**

David Klappholz, *Stevens Institute of Technology*

Vicki Almstrum, *The University of Texas at Austin*

John Clement, *St. Pius X High School, Houston, TX*

Steven Condly, *University of Central Florida*

Peter B. Henderson, *Butler University*

How often do instructors walk out of a class thinking "I really taught well today" only to discover that many of the key points escaped students' understanding? As educators, our perceptions of how we teach and what students learn are often very different. In many STEM disciplines, assessment instruments have demonstrated that even when students perform well on tests, they have major misconceptions regarding fundamental concepts. This birds-of-a-feather will be a forum for computing educators to discuss this phenomenon and ways of realistically assessing student performance. In addition, we will discuss techniques for improving instruction to counter these effects.

### **Toward a Common Automated Grading Platform**

Stephen H. Edwards, *Virginia Tech*

Bill Pugh, *University of Maryland*

The purpose of this BOF is to bring together parties interested in identifying the major roadblocks to fielding a generally-adoptable automated grading platform that is flexible enough for use across a wide variety of institutions. A platform providing a modular, plug-in-based architecture could provide a common base for advancing the interests of many working in this area, as well as a medium for infusing best practices across many institutions. In addition to brainstorming the obstacles, the group will also propose and discuss potential solutions.

## **The ACM Java Task Force API: Experiences from CS1 and CS2**

Ariel Ortiz Ramirez, *ITESM Campus Estado de México*

The Java Task Force (JTF) has done a great service to the academic community by providing a simplified way for novices to confront many of the inherent problems associated with the Java programming language. The Task Force's proposed API covers issues such as input/output, object oriented graphics, and program life cycle management. In this BOF, participants will be able to share their experiences as early adopters of the JTF API in their CS1 and CS2 courses.

## **Professional Plagiarism, Self-Plagiarism, and Computing Faculty Members**

Don Gotterbarn, *East Tennessee State University*

John Impagliazzo, *Hofstra University*

Keith Miller, *University of Illinois – Springfield*

The purpose of this session is to make participants aware of the recent events surrounding plagiarism and self-plagiarism among computing professionals and to discuss ways to address the issue. Most computing faculty members publish the results of their research. However, the way one uses information from others or from oneself may be problematic. Recently, several high profile incidents of plagiarism in computing publications have emphasized the importance of this issue. Additionally, the ACM Publications Board is developing new guidelines about the topic.

## **Unusual Technologies in Introductory Computer Science Education: From MIDI to Mobots**

Tom Lauwers, and Illah Nourbakhsh, *Carnegie Mellon University*

Zach Dodds, *Harvey Mudd College*

The majority of CS1 courses are taught using essentially the same technologies that existed fifteen years ago - a computer, programming language and some compiler/environment. In contrast, the nature and distribution of computing has changed significantly in that time; many people carry at least two portable computing devices on them; computers' capabilities have grown significantly, and are further extended by a number of interesting peripherals and the internet. A growing number of instructors view these technologies as an important avenue to connect concepts and lessons taught in CS1 to the real world. We seek to create a discussion between instructors of CS1 courses using these 'unusual' technologies, instructors interested in using such technologies, and technology developers.

## **CIS20xx/ACS20xx: Is there a need for another CC20xx Guideline?**

Ken Surendran and Dana Schwieger, *Southeast Missouri State University*

Rathika Rajaravivarma, *Central Connecticut State University*

Frank Young, *Rose-Hulman Institute of Technology*

Some Computer Science Departments offer Computer Information Systems programs (or Applied Computer Science) in addition to the traditional Computer Science. Departments in the business colleges offer Management Information Systems programs and/or Computer Information Systems. The structure of the Computing Curricula as portrayed in CC2004 Overview Reports provides for inclusion of additional programs. The purpose of this session is to assess the need for a CIS20xx or ACS20xx curriculum and, if there is sufficient support, make plans to take this initiative to the next stage.

## **The Company Store? Using Course Management Systems in Computer Science Courses**

Jeffrey L Popyack, *Drexel University*

Course management systems (CMS) are increasingly used in higher education. The typical CMS is usable by a wide audience of educators, not only those with sophisticated computing expertise. However, our experience has been that computer scientists have specialized needs and perspectives on CMS use that differ from those of the typical academic users. This discussion will focus on the possibilities and limitations of CMS systems from the perspective of computer science educators.

## **Mathematical Reasoning in Computer Science**

Peter Henderson, *Butler University*

Judith Gersting, *University of Hawaii at Hilo*

Mathematics and mathematical reasoning are central to computer science, and mathematical concepts should become an integral part of the entire CS curriculum. This birds-of-a-feather session will be a forum for educators to discuss the role of mathematics in computer science curricula.

### **Women in Computing Education Oral History Project**

Lecia J. Barker, *University of Colorado, National Center for Women and IT*

Vicki L. Almstrum, *The University of Texas at Austin*

Barbara Boucher Owens, *Southwestern University*

This BOF is designed for those who wish to find out more about the project and ways to get involved with it. The project is an extension of a Working Group at ITiCSE 2005. At the end of the session, audience members should understand the goals of the oral history project and have a basic grasp of the fundamentals of conducting and preserving interviews. The session is intended to provide the information needed for someone to decide whether they would attend a training workshop to become an interviewer. The session should also result in refinement of selection criteria and names of women who are candidate interviewees.

### **Teaching Track Faculty in CS**

Thomas J. Cortina and Don Slater, *Carnegie Mellon University*

Daniel D. Garcia, *University of California, Berkeley*

A panel at SIGCSE 2004 illustrated the trend in computer science departments to hire faculty to teach in a teaching-track position that parallels the standard tenure-track position, providing the possibility of promotion, longer-term contracts, and higher pay for excellence in teaching and service. This birds-of-a-feather is designed to gather educators who are currently in such a position to share experiences on how their work is evaluated and how they are treated as members of the faculty of their departments and schools.

### **The SIGCSE Committee on Teaching Computer Science Research Methods (CSRM)**

Anne Applin, *Ithaca College*

Hilary Holz, *California State University, East Bay*

This SIGCSE committee was formed to explore existing undergraduate and graduate courses dedicated to teaching research methods in the context of computer science as well as purposeful integration of research methods into existing courses across the curriculum where dedicated courses are not feasible. The committee provides a structure for collaborative exploration of this topic. The particular focus for this meeting of SIGCSE-CSRM is the working group we have proposed for ITiCSE 2006. Please come share your experiences and lend your help.

### **Using Torque and other Game Engines in the Classroom**

Michael P. Rogers, *Millikin University*

The use of a game engine provides students with the ability to write serious games in a semester, and can lead to a meaningful educational experience. Since most instructors grew up at a time when Game Development was not on the menu, it might be beneficial for those of us who are attempting to bring game engines into the classroom to share our experiences, good, bad, and ugly. Torque is perhaps the most popular game engine at the present time, and the game engine with which I have the most expertise: but it would be interesting to hear what other instructors are using.

### **An Introduction to the SIGCSE Committee on Models for the Evaluation of Faculty Scholarship**

Gerald Kruse, *Juniata College*

Shmuel Fink, *Touro College*

Keith Olson, *Utah Valley State College*

The goal of the proposed new SIGCSE committee is to gather resources for individuals, departments, and institutions regarding the evaluation of scholarship for computing faculty. In particular, the committee will seek to identify or draft several models for this evaluation. This BOF session will be an opportunity for the SIGCSE community to discuss the topic and provide input to the committee.

### **Are CS1 Labs Useless?**

Christopher Starr, Renee McCauley, Walter Pharr, and George Pothering *College of Charleston*

The College of Charleston recently completed a rigorous study indicating that there was no overall improvement in learning outcomes for CS1 students when closed labs were taught by the same instructor versus different instructors. We would now like to investigate whether using closed laboratories (versus no closed labs) enhance learning outcomes in CS1, which is a widely held but under-investigated assumption. The purpose of the BOF is to bring together people interested in the topic to share their experiences as well as to investigate participant interest in a multi-institutional study of CS1 with and without closed laboratories.

### **Formation of an ACM SIG on Information Systems Education**

Paul M. Leidig, *Grand Valley State University*

David L. Feinstein, *University of South Alabama*

The ACM SIGCSE Symposium has provided a forum for Information Systems educators to gather, present papers, and network along with their Computer Science colleagues. Recently, the formation of a SIG on Information Technology Education (SIGITE) has prompted the discussion of the possible formation of a SIG for Information Systems Education. Join this discussion on I.S. topics including curriculum standards and accreditation, and provide input on the process of forming a SIGISE.

### **Computer Security Education: the present, past and future**

Carol Taylor, *Idaho State University*

Rose Shumba, *Indiana University of Pennsylvania*

James Walden, *Northern Kentucky University*

Computer security became a tangible Computer Science sub discipline in the 1970's as the need to protect information became important with growing computer use in government and industry. Today, 35 years later, computer security is well established area of research and study within Computer Science. We expect to share experiences on the progress made in computer security education in the past eight years, the present state of computer security education, the goals and future objectives for security programs, and the challenges and barriers for achieving the goals.

### **Education, Computers and Society: SIGCAS**

Joseph D. Oldham, *Centre College*

Florence Appel, *Saint Xavier University*

This session presents an opportunity for several constituencies interested in computers and society to get together. SIGCSE members know the importance of societal issues in the CS curriculum. SIGCAS (Computers and Society) members are similarly committed to investigating such issues. Several SIGCSE members who hold a joint membership in SIGCAS are interested in discussing approaches to teaching societal issues in computing, and in looking for points of connection between the two SIGs.

### **Incorporating HCI into Undergraduate CS and IS Curricula**

Bill Manaris, *College of Charleston*

Mary J. Granger, *George Washington University*

Craig Miller, *DePaul University*

David R. Naugler, *Southeast Missouri State University*

Starting with SIGCSE 2002, 26 CS/IS educators have worked on building a tight yet diversified community of individuals interested in incorporating HCI into undergraduate CS/IS curricula. This has been carried out through SIGCSE activities and an NSF-sponsored workshop. Through these activities, participants have exchanged ideas and a number of new courses have appeared as a result. What is next? HCI is a dynamic area and with ever changing needs for curriculum development. This BOF will provide a forum for discussing issues related to integrating HCI into CS/IS curricula: what, where and how much?

## **Practical Planning and Implementation Strategies for a new CS Sequence, Concentration, or Degree Program**

Chu J. Jong and Terry Dennis, *Illinois State University*

As the demand for graduates possessing the latest technological skills grows, there is increased pressure on educational institutions to offer new course offerings and sometimes even whole degree programs that address these needs. Courses in wireless communication, information security, web services, web hosting, data-mining services, and new computer system architectures are examples of such courses and/or programs. Obviously, short-lived changes do not lend themselves to new concentrations or majors. We will offer the development of our new degree program, Enterprise Systems Computing, as an example to initiate the discussion.

## **Price Inflation of CS Texts?**

Deepak Kumar, *Bryn Mawr College*

The average price of the 8-10 top-selling CS1 texts is in the \$92-\$105 range (the three top-selling titles average \$98) whereas the price of introductory Java or C++ books that are not designated as texts average \$30-\$40. Moreover, many texts that include software have licensing issues when ownership is transferred because of a resale. All of this results in higher costs to students and issues with licensing and reselling. This BOF will bring together SIGCSE attendees, authors, and representatives from publishers to discuss and perhaps address these issues.

## **Experiential Learning in the CS/IS Curriculum**

Jeffrey A. Stone and Elinor Madigan, *Pennsylvania State University – Schuylkill*

Experiential learning can be a powerful tool in the CS/IS curriculum. Experiences like internships, co-curricular activities, lab exercises, and external projects are effective tools for active learning and provide students with a means to connect their academic work and its place in society and/or the workplace. The reflection mechanism inherent in experiential learning allows students to translate their actual experience into a form that has lasting meaning. In this session we will explore different methods for offering experiential learning experiences for students, both in and out of the classroom. Best practices, lessons learned, and innovative pedagogy will be discussed.

## **Recruit and Retain: Innovative changes to computer science education for the recruitment and retention of women**

Tanya L. Crenshaw, *University of Illinois at Urbana-Champaign*

Heather Metcalf, *University of Arizona (Women's Studies)*

This BOF discussion will focus on new ideas, existing methods, and changes to existing computer science education that can help to improve the recruitment and retention of women in this field at three levels of education: high school, college, and graduate school.

# SIGCSE 2006 Faculty Poster Sessions

Posters present work-in-progress and other topics for which dialog with Symposium attendees is particularly appropriate. Posters are on display Friday morning and afternoon, and authors will be available for discussion during these sessions.

## Session I: Friday, March 3, 9:45 – 11:45, Ballroom of the Americas

### **Objects First, Design Patterns Second: Lessons Learned from the `Killer Examples' for Design**

Carl G. Alphonse and Adrienne Decker, *University at Buffalo*  
Michael E. Caspersen, *University of Aarhus*  
Martha Kosa, *Tennessee Technological University*  
Stephen Wong, *Rice University*

### **Logic Demonstrator**

Richard D. Barnhart and Richard D. Koontz, *Grace College*

### **Introducing Software Engineering Principles in CS0 Using RoboLab™**

Jeremy Cannell and Stephen Frezza, *Gannon University*

### **CS Outside `the Box:' Assignments and Resources for Computing with External Devices**

Zachary Dodds, *Harvey Mudd College*

### **Teaching Web Services with Water**

Edward F. Gehringer, *North Carolina State University*

### **An Innovative Approach in Teaching Object-oriented Programming to the Novice**

Ahmad Ghafarian, *North Georgia College & State University*

### **Recapturing the Interest of Females in Computer Science at Norfolk State University**

Jonathan M. Graham, Cheryl V. Hinds and Mary L. Williams, *Norfolk State University*

### **Linking Mathematics and Computer Science: A Summer Workshop Experience for 6-12 Grade Teachers**

Mindy L. Hart, *Purdue University*

### **Eclipse and Collaborative Tools for Group Projects**

Ken T. N. Hartness, *Sam Houston State University*

### **Non-traditional Doctor of Science in Applied Computing, with Integrated Distance Technology Support**

Kenneth Hoganson, *Kennesaw State University*

### **Computer Experience and the Computer Science Student**

Meriel Huggard, *Trinity College, Dublin, Ireland*

### **A Two- (and a Half) Pass Assembly Language Course**

Scott B. Hunter, *Siena College*

### **Parallel Java: An API for Teaching and Developing Parallel Programs in 100% Java**

Alan Kaminsky, *Rochester Institute of Technology*

### **Reaching Out to Non-Computer Science Majors**

Patricia H. Lapczynski and Anthony Serino, *College Misericordia*

### **Encouraging Women to Enter Hi-tech Professions**

Meirav Mano and Bruria Haberman, *Holon Academic Institute of Technology*

### **Objects First Using Games and Lego Robots: A Tri-P-LETS Initiative**

Lee McCauley, Linda B. Sherrell, David Mills, Jim Greer, Quynh Tran and Sarah Bowen, *University of Memphis*

**Development of a Graduate-Level Digital Forensics Curriculum**  
Timothy J. McGuire and Karon N. Murff, *Sam Houston State University*

**Discrete Mathematics with Objects**  
Kirby McMaster, Ronald Peterson and Nicole Anderson, *Weber State University*

**A Survey of Introductory Computer Science Course Instructors**  
Illah Nourbakhsh and Tom Lauwers, *Carnegie Mellon NSH*

**A Targeted Tablet PC Software Development Course**  
Roy Pargas, *Clemson University*

**Integrating Empirical Methods into the Computer Science Curriculum**  
David Reed, *Creighton University*  
Craig S. Miller, *DePaul University*  
Grant Braught, *Dickinson College*

**Injecting Offshore Outsourcing into the CS/IS Curriculum**  
Christelle Scharff, Olly Gotel, and Imtiaz Amzad, *Pace University*  
Sopheap Seng, *Institute of Technology of Cambodia*

**Being Simple: The Use of a Machine Language Simulator to Teach CS1 Concepts**  
Jeffrey A. Stone, *Pennsylvania State University*

**Sustaining Learning in Early Major Computer Science Courses**  
Vinayak Tanksale, *Ball State University*

**Once She Makes It, She Is There: Gender Differences in Computer Science Study**  
Tamar Vilner and Ela Zur, *Open University of Israel*

## **Session II: Friday, March 3, 3:15 - 5:15, Ballroom of the Americas**

**Green: An Eclipse Plug-in to Support Teaching Design Issues in CS1-CS2 Using Round-tripping**  
Carl G. Alphonse, *University at Buffalo*

**A Project-based Course in Small-scale Embedded Systems**  
Kevin B. Austin, *Fitchburg State College*

**Examples of WeBWork Programming Assignments**  
Jackie Baldwin and Andrew Wildenberg, *Cornell College*  
Eileen Crupi, Tabitha Estrallado, Olly Gotel, Richard Kline and Christelle Scharff, *Pace University*

**High School Students Studying Towards an Undergraduate Degree in Computer Science**  
Tamar Benaya and Ela Zur, *Open University of Israel*

**Curriculum Material for Teaching Digital Media as Computer Science**  
Jennifer Burg, *Wake Forest University*

**PDA-based Interaction Support in Large Lecture Venues**  
Henning Bär and Guido Rößling, *Darmstadt University of Technology*

**Bridging Undergraduate Learning and Research in Software and Hardware via Network Systems Design Using Network Processors**  
Liang Cheng, *Lehigh University*

**So You Want Your Students to Design Educational Applications? Learning Technology Design as a Context for Computer Science Project Work**  
Chris DiGiano, *SRI International*

**Retaining Underrepresented Groups in First Year CS Courses**

Sandra DeLoatch, *Norfolk State University*  
Aurelia T. Williams, *Norfolk State University*  
Nikki Jackson, *Norfolk State University*

**Expertiza: Reusable Learning Objects through Peer Review**

Edward F. Gehringer, *North Carolina State University*

**Strategies to Incorporate a Database Courseware into Introductory Database Classes**

Mario Guimaraes, *Kennesaw State University*

**Students' Difficulties with Abstraction and Code Reuse in the Declarative Logic Programming Environment**

Bruria Haberman, *Holon Academic Institute of Technology*  
Zahava Scherz, *The Weizmann Institute of Science*

**Student Attitudes Toward Pair Programming**

Brian Hanks, *Fort Lewis College*

**How I Taught a Semester of Computer Graphics in Two Weeks, in Germany, and Lived to Tell about It...**

Gerald Kruse, *Juniata College*

**A Task-oriented and Goal-driven Approach**

Jigang Liu, *Metropolitan State University*  
Kai Qian, *Southern Polytechnic State University*

**Mini-Projects for an Advanced Database Course**

Kirby McMaster, Nicole Anderson and Ashley Blake, *Weber State University*

**Computer Science Students' Difficulties with Proofs by Induction: A Study**

Ana Pasztor and Irene Polycarpou, *Florida International University*

**The MMIX Motherboard**

Martin Ruckert and Axel Böttcher, *Munich University of Applied Sciences*

**Fostering Undergraduate Research Using Evolutionary Computation**

Bart Rylander, *University of Portland*

**Secure Software Engineering: A Model for a Software Project Course**

Rose Shumba, *Indiana University of Pennsylvania*  
James Walden, *Northern Kentucky University*

**Integrating Secure Coding Into a Computer Science Curriculum**

Carol Taylor, *University of Idaho*

**Integrated Research Components: Ensuring That All Students Can Have a Research Experience**

Ellen L. Walker and Oberta A. Slotterbeck, *Hiram College*

**Completing the Circle: Designing a Senior Project Capstone Course**

Scott Weiss, *Mt. St. Mary's University*

**A Semi-Moore Method Approach to Teaching Advanced Programming**

Dorian Yeager, *Grove City College*

**PSP Log Tool for Teaching the Personal Software Process**

Wook-Sung Yoo and Stephen Frezza, *Gannon University*

# SIGCSE 2006 SRC Graduate Student Research Abstracts

## **A Study on the Cognitive Skills Involved in Computer Programming**

Jorge Vasconcelos, *Johns Hopkins University*

An ongoing study exploring cognitive skills associated to computer programming is presented. A novel approach involving surveying problem solving abilities (weighting wrong answers rather than correct ones), and developing a taxonomy of thinking skills (like “big picture”, “black box” and “state machine” thinking) essential to the field are introduced. This knowledge can help to get a better perspective of the reasons for students struggle during programming courses, as well as develop indicators to have better assessment techniques for cognitive development rather than skill proficiency.

## **Value Driven System Level Test Case Prioritization**

Hema Sirkanth, *North Carolina State University*

Companies have limited resources that restrict their ability to effectively complete testing efforts. I build upon prior research to present system-level test case prioritization scheme called the Prioritization of Requirements towards Testing (PORT). PORT involves analyzing and assigning value to each requirement for four factors: requirements volatility, customer priority, implementation complexity, and fault proneness. System test cases are prioritized for execution based on the assigned priority of the requirements they originate from such that the test cases for requirements with higher priority are executed earlier. PORT was applied on two studies to show improved rate of detection of severe failures.

## **Fast Fourier Transforms on High Performance Computing Systems**

Purvi Shah, *University of Houston*

Fast Fourier Transform (FFT) is one of the most widely used algorithms in science and engineering. However efficient FFT implementations for execution on High Performance Computing (HPC) systems are difficult to build due to the growing heterogeneity of hardware architectures, complexity of memory systems and diversity of communication systems of HPC platforms. The UHFFT is an adaptive, portable, efficient and easy-to-use library. This paper describes the automatic performance tuning implemented in the UHFFT library.

## **A Systematic Approach for Separation of Concerns in Evolving Software Systems**

Haitham Hamza, *University of Nebraska-Lincoln*

This research addresses the problem of modularize software systems into units that can easily evolve when changes occur to the system. We argue that, existing software modularization techniques do not explicitly consider system evolution when decomposing the system, and hence, adapting the system may trigger invasive changes, leading to expensive and complex evolution process. In addition, there exist no systematic techniques for decomposing software into evolving modules. This research attempts to address these problems by developing a systematic modularization approach to decompose systems into stable modules that can evolve over time while preserving their structure.

# SIGCSE 2006 SRC Undergraduate Student Research Abstracts

## **Using Dynamic Time Warping and a Variation to Predict Perceived User Quality of Media Streams**

Jamie Olson, *Carleton College*

An important step towards improving the quality of multimedia streaming is the development of a universal, robust quality measurement. We test the use of dynamic time warping to predict user-perceived quality. Dynamic time warping is compared with a variety of other data mining techniques for use in predicting network loss and user rankings.

## **Adapting K-medians Clustering for Normalized Cluster Centers**

Anna Ritz, *Carleton College*

With increasing pollution levels, atmospheric chemistry is a prominent and relevant area of research. Atmospheric particle data is difficult to analyze because it is high-dimensional, sparse, and normalized. The dataset size is also often an obstacle, so clustering techniques are important for identifying trends in the data. Although the popular k-means algorithm has been adapted to produce optimal normalized cluster centers, no similar adaptation has been done for the k-medians clustering algorithm. We present a new adaptation to the k-medians clustering algorithm that produces locally optimal normalized cluster centers.

## **Community-Based World Wide Web Content Filtering**

Jason Schramm, *The College of New Jersey*

The Community-Based World Wide Web (WWW) Content Filter is a system to identify inappropriate content on the World Wide Web using Bayesian statistics and community input. The approach is similar to spam filtering, but is applied on a wider scale and with an ambitious goal to improve the quality of the information we see everyday on the Internet. The community determines what content is appropriate, and the probability of a page being bad is calculated based on data collected from sites that are determined by the community to be good or bad.

## **The Implementation of an Ad-hoc Environment for Sharing a Fixed Resource with "rMusic"**

Eric Thul, *The College of New Jersey*

The "rMusic" project focuses on the sharing of a fixed resource, which in this case, is a set of speakers. We want to find out how to effectively communicate data within a peer-to-peer network and manage dynamic input from hand-held devices, in order to allow for the creation of an ad-hoc environment in which users may establish a sense of credibility. This is accomplished by implementing a peer-to-peer system allowing for dynamic play-list generation based on user influence. By investigating this phenomenon, we can possibly learn an effective means to share a fixed resource within a dynamic community without an administrator.

## **Unsupervised Thresholding and Morphological Processing for Automatic Fin-outline Extraction**

Scott Hale, *Eckerd College*

Software packages exist to identify cetaceans—whales, dolphins, porpoises—, which can be uniquely identified based on the features of their dorsal fins. These packages, however, require much user interaction/effort to extract a fin outline from a photo. This research aims to provide an automated method (employing unsupervised thresholding techniques and image manipulation methods) to extract a cetacean dorsal fin outline from a digital photo so that manual user input is reduced and overall user experience improved in such packages.

## **Software Optimizations in a Microcontroller-based Environment**

Matthew Glover, *Fitchburg State College*

When developing software for a microcontroller, applications are constrained by memory limitations. After application development has exhausted memory resources and functionality has yet to be achieved the code requires optimization. Optimization, in our case, is a source code modification to obtain maximum efficiency necessary to allow the addition of new functionality. Using tools included with our IDE, such as the statistics generator, C/ASM list, and the call tree, code for the "Virtual Firefly" was optimized to allow for the addition of key functions to our microcontroller software as application requirements changed.

## **Firefly Flash Emulator for Testing of Virtual Firefly Device**

David Lustig, *Fitchburg State College*

With the goal of creating tests that accurately assess firefly-flash-logging hardware, it was found that precise control over ambient light and the accurately synthesized flash of a firefly was needed. To meet this need, the Firefly Flash Emulator was designed from the bottom up, based on the simple yet efficient combination of the pulse width modulator with an RC filter. Through a straightforward keyboard and LCD interface, any hardware tester now has the freedom to set the rise, sustain, and fall times for a fabricated flash, as well as its intensity and surrounding ambient light levels.

## **Locating, Tracking, and Interpreting EAN-13 Bar Code Waveforms in a Two-dimensional Video Stream**

Jeffrey Adair, *Hiram College*

A robust method of locating and interpreting bar codes in a two-dimensional video stream is discussed. Using a variation of the hough transformation allows the scanner to use all pixel data associated with the bar code to determine the scan waveform. This creates a system that is better able to handle bar code data which is noisy, or damaged. This results in a scanning system that produces usable data more frequently than traditional methods.

## **Programmer-Defined Formatting**

Megan Patnott, *Hope College*

The source code for computer programs is notoriously hard for people to read and understand. This adds to the difficulty of maintenance, which is a large part of the life cycle of most programs. Programmer-defined formatting allows programmers to use formatting options, such as font size, style, color, and weight, in a non-algorithmic way in order to better communicate its meaning. The programmer-defined formatting can then act as an aid in reading and understanding the code.

## **Inter-Disciplinary Undergraduate UAV Project**

Brent Bradbury, *LeTourneau University*

The project is an inter-disciplinary, multi-year, undergraduate unmanned autonomous vehicle project incorporating teams from both engineering and computer science departments. The computer science team designed and implemented a guidance and sensor management system, wrapping telemetry and PID-based flight control into a single unit. A COBRA interface communicates sensor information throughout the physical system. Testing of guidance and flight control models has been performed virtually using a plug-in to the X-Plane simulator.

## **An Evaluation of the Edit-Distance-With-Moves Similarity Metric for Comparing Genetic Sequences**

Shiri Azenkot, *Pomona College*

We describe the first known implementation of an approximation algorithm for the string edit distance with moves similarity metric. This is the first algorithm to consider nontrivial alignment and run in substantially sub-quadratic time [CM02]. Extensive experimentation demonstrates that the algorithm produces a good approximation for the edit distance with moves metric, especially on strings of length 500B to 10KB. We also found that the algorithm has high potential for use in computational biology. When comparing texts of genetic sequences, our algorithm outperforms the q-grams heuristic in predicting results of the Smith-Waterman algorithm. Finally, we propose additional application areas for our implementation.

## **Tree Visualization in JAWAA**

Andrea Gibson, *University of Colorado – Boulder*

Being able to visualize algorithms and data structures is an important part of learning computer science. JAWAA is a pre-existing tool to do just that. Our goal was to improve JAWAA to automatically handle tree layouts, and improve other aspects of it to make it more visually pleasing, as well as provide greater capability to demonstrate various algorithms and data structures.

## Author/Panelist Index

Adams, J. C. .... 359	Constantine, M. G. .... 264	Gal-Ezer, J. .... 389
Agarwal, R. .... 259	Cooper, S. .... 560	Gehring, E. F. .... 458
Allen, D. A. .... 378	Cornwell, T. .... 428	George, B. .... 7
Almstrum, V. L. .... 195	Cortina, T. J. .... 67	Ginat, D. .... 469
Al-Muhtadi, J. .... 443	Cross II, J. H. .... 131, 456	Goldman, K. J. .... 560
Alt, C. .... 438	Cunningham, S. .... 266	Goldweber, M. .... 61
Amoussou, G.-A. .... 558	Cutler, R. .... 131, 201	Gonzalez, G. .... 133
Anderson, R. .... 344		Gousie, M. B. .... 72
Anewalt, K. .... 209	Dansdill, T. .... 117	Granger, M. J. .... 558
Angel, E. .... 266	Davies, G. .... 456	Gray, K. E. .... 288
Applin, A. G. .... 530	Davis, D. .... 339	Grissom, S. .... 131
Armoni, M. .... 197, 389, 469	Davis, G. .... 201	Griswold, G. .... 219
Aspray, W. .... 330	Decker, A. .... 46	Gross, P. .... 560
Astrachan, O. .... 438	DeHart, J. .... 107	Guzdial, M. .... 396
	Deibel, K. .... 63	
Barnes, T. .... 394	Deibel, K. .... 458	Hahne, R. .... 378
Bayliss, J. D. .... 500	Denning, P. .... 65	Hamer, J. .... 458
Benham, D. .... 332, 453	Denning, T. .... 219	Hankins, J. .... 373
Berman, K. A. .... 464	Denton, L. F. .... 138	Hazzan, O. .... 22, 408
Bierre, K. .... 354	DePasquale, P. J. .... 214, 283	Henry, S. .... 234
Blahnik, J. .... 191	Deremer, D. .... 179	Herbert, K. G. .... 179
Blum, L. .... 22	DeRosa, P. .... 308	Herscovici, D. S. .... 117
Borunda, P. .... 244	Dershem, H. .... 339	Hoffman, M. E. .... 117
Bower, M. .... 148	Dias, M. B. .... 22	Horstmann, C. .... 378
Bressler, B. .... 379	Dodds, Z. .... 448	Horwood, J. .... 158
Brewer, C. .... 244	Dubinsky, Y. .... 408	
Bruce, K. .... 131	Duvall, R. C. .... 224	Impagliazzo, J. .... 456
Budd, T. A. .... 143		Israel, D. .... 443
Bullers, Jr., W. I. .... 102	Eckerdal, A. .... 403	
Burd, S. .... 102	Edwards, S. H. .... 259	Janzen, D. S. .... 254
Byckling, P. .... 413	Egan, M. A. L. .... 63	Jipping, M. J. .... 234
	Egert, C. .... 46, 354	John, D. J. .... 185
Cain, J. .... 562	Eidelman, L. .... 408	
Callele, D. .... 433	Epstein, R. G. .... 535	Kamali, R. .... 456
Canosa, R. L. .... 543	Erten, C. .... 244	Kamin, S. .... 443
Carlisle, M. .... 560	Etlinger, H. A. .... 122	Karp, L. .... 448
Carter, L. .... 27		Kazemzadeh, M. B. .... 510
Cassel, L. .... 199	Fagin, B. .... 61	King, N. .... 244
Chamillard, A. T. .... 515	Ferguson, R. .... 127	Klappholz, D. .... 195
Chen, T.-Y. .... 495	Fetrow, J. S. .... 185	Klassner, F. .... 61
Christensen, K. .... 229	Finley, T. .... 379	Klee, K. .... 131
Clement, J. .... 195	Forbes, J. .... 438	Knox, D. L. .... 214
Cliburn, D. C. .... 77	Frens, J. D. .... 92	Konidaris, G. .... 398
Cohen, R. F. .... 63, 279	Frieze, C. .... 22	Kontostathis, A. .... 553
Condly, S. .... 195		Kortsarts, Y. .... 460

Kouril, M. ....	464	Miliszewska, I. ....	158	Rauchas, S. ....	398
Krishna Rao, M. R. K. ....	418	Miller, B. ....	396	Razmov, V. ....	344
Kuhn, S. ....	525	Monge, A. ....	495	Reading, S. ....	379
Kuhns, F. ....	107	Moström, J. E. ....	403	Reed, D. ....	378
Kumar, A. ....	479	Motil, J. ....	562	Reges, S. ....	293, 562
Kumar, D. ....	479	Murphy, L. ....	17	Reichgelt, H. ....	129
		Musicant, D. R. ....	538	Robbins, S. ....	303
La Barge, B. ....	564			Roberts, E. ....	131, 334, 562
Labrador, M. A. ....	558	Naps, T. ....	328	Roden, T. ....	510
Lance, C. ....	373	Nation, Z. ....	244	Rodger, S. ....	131, 197, 328, 438
Larson, E. ....	249	Nevison, C. ....	562	Rodger, S. H. ....	379
Last, M. ....	127	Niezgoda, S. ....	51	Rosenbloom, P. ....	65
Layman, L. ....	428	Noonan, R. E. ....	51	Rosmaita, B. J. ....	63, 270
LeBlanc, M. D. ....	313			Rosman, B. ....	398
LeBlanc, R. ....	456	O'Leary, M. ....	2	Ross, R. ....	328
Leibowitz, R. ....	313	Osborne, L. J. ....	112	Rößling, G. ....	328
Lent, R. ....	264			Roumani, H. ....	491
Levine, D. B. ....	173	Pankratz, D. ....	191	Rufinus, J. ....	460
Lewis, M. C. ....	505	Parberry, I. ....	394, 510	Rundus, D. ....	229
Liao, W. ....	12	Pargas, R. P. ....	474	Russell, I. ....	479
Liu, C. ....	127, 423	Parlante, N. ....	562		
Lopez, F. G. ....	264	Parwatikar, J. ....	107	Saiedian, H. ....	254
Lopez, Jr., A. M. ....	264	Pattis, R. ....	1	Sajaniemi, J. ....	413
Lucic, R. ....	438	Paul, J. L. ....	464	Sakhnini, V. ....	408
Ludewig, K. ....	234	Payne, T. ....	169	Sanders, I. ....	398
Lunt, B. ....	456	Pearson, J. ....	308	Sanderson, P. ....	239
		Pedroni, M. ....	481	Scheessele, M. R. ....	548
Makaroff, D. ....	433	Pengelley, D. ....	318	Schlesinger, R. ....	332
Marion, B. ....	268	Perera, G. ....	229	Schriefer, T. ....	548
Martin, F. G. ....	82, 525	Pérez-Quñones, M. A. ....	259	Seazzu, A. F. ....	102
Massingill, B. ....	505	Perry, S. ....	558	Serino, C. ....	349
Matsuura, S. ....	163	Peterson, D. L. ....	443	Shackelford, R. ....	456
Mayadas, A. F. ....	330	Peterson, L. I. ....	332, 453	Shah, D. M. ....	474
McCann, L. I. ....	562	Phelps, A. ....	354	Shen, K. ....	308
McCartnery, R. ....	403	Pivkina, I. ....	318	Shirley, P. ....	266
McCauley, R. ....	17	Polack-Wahl, J. ....	209	Shokhirev, M. ....	244
McFall, R. ....	339	Pollard, S. ....	224	Shurn, T. ....	364
McGettrick, A. ....	65	Polze, A. ....	298	Siegfried, R. M. ....	275
McGettrick, A. ....	199, 456	Powers, K. ....	560	Simon, B. ....	219, 495
McKenna, R. ....	67	Probert, D. ....	298	Skaff, J. ....	279
McKinney, D. ....	138	Proulx, V. K. ....	288, 560	Sloan, R. ....	199, 456
McNally, M. ....	61, 560	Pulimood, S. M. ....	214	Smith, S. ....	349
McVey, B. ....	191			Snyder, L. ....	65
Meacham, A. ....	279	Quinn, M. J. ....	520	Soh, L.-K. ....	486
Mead, J. ....	32			Standley, H. M. ....	37
Meneely, A. ....	92	Ranjan, D. ....	318	Stephenson, C. ....	201
Mertz, J. ....	127	Ranum, D. ....	396	Stevenson, D. E. ....	41
Meyer, B. ....	481	Ratcliffe, M. ....	403	Stewart, C. ....	308

Stoecklin, S. ....	349	Villaverde, K. ....	318	Wong, K. ....	107
Strout, S. ....	500	Vollmar, K. ....	239	Woszczyński, A. B. ....	153, 332
Sung, K. ....	266				
Tableman, L. ....	234	Waegel, D. B. ....	553	Xu, J. ....	12
Talton, J. O. ....	443	Wagner, P. J. ....	323	Xu, L. ....	82
Teif, M. ....	408	Waite, W. M. ....	87		
Topi, H. ....	456	Walker, H. M. ....	173	Yang, T. A. ....	97
Tran, S. P. M. ....	97	Way, T. P. ....	51, 203, 384	Yaverbaum, G. ....	129
Trees, F. ....	131	Westbrook, S. ....	17	Yellin, F. ....	131
Turner, J. ....	107	White, T. M. ....	384	Yoo, J. ....	373
		Whittington, K. J. ....	458	Yoo, S. ....	373
Utting, I. ....	131	Wick, M. R. ....	41, 323, 394	Yu, H. ....	12
		Wilkerson, M. ....	219	Yuan, X. ....	12
Valeva, A. ....	7	Williams, L. ....	428		
Van Busum, K. M. ....	558	Winters, T. ....	169	Zander, C. ....	403
Vardi, M. ....	197, 330	Wiseman, C. ....	107	Zelle, J. ....	396
Ventura, P. ....	46, 354	Wolfman, S. A. ....	562	Zulli, S. ....	229
Verma, R. ....	197	Wolz, U. ....	394	Zweben, S. H. ....	129, 330

