

SIGCSE 2009 Workshops

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

1. Greenfoot - Introducing Java With Games And Simulations

Michael Kölling, *University of Kent*

Greenfoot is a programming environment, developed by the creators of BlueJ, that allows teaching of object-oriented programming concepts – using Java – in a highly engaging and motivating context. Built in the tradition of Turtle Graphics, Karel the Robot and Squeak, Greenfoot offers a more integrated, graphical, interactive experience than previous systems. Building widely differing scenarios, such a simulations or games, is easy and quick, and scenarios can easily be published on a website (greenfootgallery.org). Greenfoot should be of interest to anyone teaching Java, especially in early programming courses (including schools). The workshop is practically oriented and allows participants to use Greenfoot in their classroom immediately. The intended audience is teachers of introductory programming courses (in Java), both at high school level and college/university level. Laptop highly recommended. Please install Greenfoot (from www.greenfoot.org) before the workshop. Participants without laptops will be paired up with a partner with a laptop.

2. Introduction to Game Design and Development

Brian Ladd, *State University of New York at Potsdam*

Tiffany Barnes, *University of North Carolina at Charlotte*

Dan Cliburn, *University of the Pacific*

Game development in computer science education motivates students, tapping into students' interests and experience. However, it is a challenge to create meaningful, pedagogically sound game assignments and courses that allow for both creativity and learning while also using modern video game design practices. We will discuss sample game development syllabi and projects from introductory to advanced computing courses, and introduce three freely-available, novice-friendly game engines. Attendees will gain hands-on experience completing game development assignments, as well as guidance on grading and managing game assignments, which will prepare them to teach an introductory course in game design and development. Laptop: Required; MS Windows w 2MB RAM Intended Audience: Computer science instructors with little or no experience using video game assignments or video game course material; instructors with an interest in using these techniques.

3. Writing more effective NSF proposals

Stephen Cooper, *National Science Foundation*

Victor Piotrowski, *National Science Foundation*

Timothy Fossum, *State University of New York at Potsdam*

This workshop will focus on writing effective education proposals. The workshop will cover the following topic areas: writing effective goals, objectives and outcomes; writing a convincing rationale; preparing a responsive evaluation plan; preparing an effective and believable dissemination plan; and dealing with the realities of the review process. The workshop will use a structured guided-interactive methodology that leads participants through each of these topics by introducing the issues related to the topic, engaging participants in group exercises designed to explore and share their understanding of these issues, and then providing some “expert” opinion on these issues. Intended audience: Faculty who are interested in writing stronger NSF education proposals. Laptop not needed.

4. Streamlining the Outcomes Assessment Process: the FCAR Methodology

John Estell, *Ohio Northern University*

This workshop presents a streamlined methodology for both course and program outcomes assessment in support of the new ABET CAC accreditation criteria for computing programs. After reviewing the new ABET program outcomes and the structured format of the Faculty Course Assessment Report (FCAR), the use of performance vectors for conducting direct student assessment will be introduced through hands-on Excel-based exercises. Assessment reports will be prepared in Word through collecting FCAR-derived information into performance vector tables, inserting these tables into the report, then applying heuristics to categorize aggregate student performance. Participants will practice evaluating assessment reports by focusing attention on areas indicated as not meeting expectations and developing action plans that facilitate continuous improvement efforts. Intended audience: those dealing with assessment/accreditation issues or who are interested in assessment methodologies. Laptop required.

5. Active and Cooperative Learning Techniques for Computer Science

Jeffrey 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. The intended audience is faculty members who are just learning about active and cooperative learning as well as those who are using these pedagogies already and are looking for additional ideas and problem solving strategies. A laptop is not needed for this workshop.

6. Teaching Computer Graphics with GLSL Shaders

Steve Cunningham, *Brown Cunningham Associates*
Mike Bailey, *Oregon State University*

Knowing how to use programmable shaders is already important and will soon become critical for computer graphics. The new OpenGL 3.0 standard replaces fixed-function operations with shaders and deprecates the traditional pipeline. We introduce vertex, fragment, and geometry shaders, discuss shader theory and the GLSL shader language, and describe teaching shaders in introductory and advanced courses. We show examples and discuss details of each class of shader. Because learning to program shaders is difficult, we introduce a tool, *glman*, that lets students develop shaders separately from applications. With *glman*, it is easy to tune shaders and experiment with their parameters. The workshop is intended for faculty who teach a computer graphics course and want to include shader concepts or for anyone who wants to learn about graphics shaders. A Windows laptop with shader-capable graphics card is recommended but not needed.

7. Web Application Security

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

The most common software vulnerabilities are found exclusively in web applications: cross-site scripting (XSS) and SQL injection. The doubling of the number of reported software vulnerabilities since 2004 is largely a result of these bugs, examples of which are taught as good coding practice in many textbooks. Participants will learn penetration testing and code review techniques and tools for finding SQL injection and XSS vulnerabilities through hands-on exercises. They will also learn to write secure code that doesn't have such vulnerabilities. The exercises will use a demonstration blogging application that participants can use in their own courses. Intended Audience: Computer science instructors who teach web application development. Participants should be familiar with developing web applications in a language like Perl, PHP, or Java, but they do not need to have prior experience with secure programming or software security. All participants must bring a laptop capable of running VMware. We recommend 1GB RAM and at least 4GB of free disk space. Laptop Required.

8. An Introduction To Version Control With Subversion

John Cigas, *Park University*

This workshop introduces participants to using Subversion, a version control package that provides support for teams of developers working on shared documents, especially source code. In the classroom, this is beneficial for students collaborating on group programming projects and for individuals to easily undo changes when they prove unfruitful. Participants will work with an established repository so they can focus on common, daily-use concepts, such as committing, updating, merging and resolving conflicts, rather than on the details of initial setup and other one-time activities. The workshop repository contains files for a web site, which is not only understandable by a wide range of participants, but also allows for easy viewing of edits and changes during the course of the workshop exercises. Intended audience: This workshop is for people new to using a version control system, or for those who have used a version control system as a single user, but not as part of a group project. Those with experience on one or more group projects under version control will probably not find the workshop interesting or helpful. Laptop Required: All participants will need a laptop to complete the exercises. Sharing is possible, though not ideal. Participants should download and install a subversion client before the workshop so they'll be ready to hit the ground running.

9. Enhancing Networking Courses with the Open Network Laboratory

Charlie Wiseman, *Washington University in St. Louis*
Ken Wong, *Washington University in St. Louis*
Jon Turner, *Washington University in St. Louis*
Tilman Wolf, *University of Massachusetts, Amherst*

The Open Network Laboratory (ONL) is an Internet-accessible virtual networking testbed that provides a sandbox environment for students to study Internet protocols and mechanisms using real networking hardware. This workshop will introduce ONL as a tool that can be used to enhance the learning experience in networking courses. We will start by covering the basics of how to use ONL through the Remote Laboratory Interface, and then give example lab assignments that utilize ONL to teach specific networking concepts. Participants will be actively using ONL to follow along during the entire workshop, so laptops with wired connections are required. ONL is open to all computer science educators and researchers. Additional information about ONL is available at <http://onl.wustl.edu>.

10. Engaging Student Learning Through Virtual World Programming

Dean Zeller, *Kent State University*
Sarah Skibiski, *Sarah Nerd Estates and Mainland Dealings*

The latest trend in computer-based education is through virtual worlds. The increasing speed, storage capacity, and networking bandwidth of personal computers have made virtual worlds a feasible communication and entertainment media. This workshop introduces participants to the educational aspects of Second Life, a popular virtual world. Participants will receive hands-on experience with Second Life, including avatar appearance, navigation, communication, finances, building objects, programming scripts, and more. Also discussed are limitations of Second Life, social controversy over virtual world education, and the competing virtual world programs. The intended audience for the workshop is anybody who would like to learn more about the uses of the Second Life virtual world for educational purposes. Programming experience is useful but not necessary for this workshop. Participants will need a wireless Internet laptop with the Second Life software installed. The Second Life software is available at <http://secondlife.com/downloads..>

11. Writing Computer Books

Barry Burd, *Drew University*
Rick Decker, *Hamilton College*

This workshop covers the computer book-writing process from start to finish. It covers both textbooks and books for the general public. Workshop topics include choosing a subject, writing a proposal, submitting sample chapters, finding a publisher, contacting an agent, reading a contract, meeting deadlines, working with coauthors, reviewing copy edited material, responding to technical reviews, creating ancillary materials, and marketing your book. The presenters are computer science professors so (naturally) the discussion will be honest, informative, and unbiased. If you've ever considered writing a book, this workshop is for you. Note: This workshop does not include a lab. Bring your laptop only if you want to play solitaire during the presentation/discussion.

12. Teaching with the GridWorld Case Study

Paul Tymann, *Rochester Institute of Technology*
Laurie White, *Mercer University*

Case studies expose students to large programs, enabling them to understand the importance of design and good programming style while encouraging teamwork and active learning. Creating a good case study that illustrates the key ideas of CS-1 is difficult and time consuming. Fortunately, there are existing case studies that can be used. This workshop will highlight the GridWorld case study that was developed to support the AP CS curriculum. The workshop will begin with an introduction to GridWorld through a series of guided activities followed by examples of how GridWorld has been used in CS-1 courses. Intended Audience: This workshop is designed for secondary and early post-secondary CS teachers who are interested in incorporating GridWorld in introductory computer science courses that use Java. Laptop Required: All participants will need a computer to actively participate in this workshop. Eclipse and Java 1.5 (or greater) must be installed on the computer.

13. Computer Security Essentials (Part 1) - System Footprinting and Vulnerability Assessment

Paul Wagner, *University of Wisconsin - Eau Claire*
Daren Bauer, *University of Wisconsin - Eau Claire*
Thomas Paine, *University of Wisconsin - Eau Claire*
Jamison Schmidt, *University of Wisconsin - Eau Claire*
Jason Wudi, *JAMF Software*

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 VMWare 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. Intended Audience: Computer Science educators interested in developing curricula in the area of computer security and information assurance. This workshop will focus on how system information is gathered by security professionals as well as attackers for the purpose of assessing system security and/or vulnerability. Laptop required. Each participant's laptop should have wireless network support.

14. Reading and Writing like Computer Scientists: How to Promote Critical Thinking and Student Engagement

Mark Hoffman, *Quinnipiac University*
Timothy Dansdill, *Quinnipiac University*

This workshop introduces participants to an informal writing process that promotes student engagement and critical thinking. Unlike formal writing typically used in software development or capstone courses to demonstrate knowledge, informal writing supports student learning (i.e., writing as thinking). Participants will use the Prioritize, Translate and Analogize (PTA) process in a model assignment; discuss how it works; and use it to develop a writing assignment. Participants will receive materials for the workshop assignment, samples of computing assignments employing the PTA process, and other support materials. Participants are encouraged to bring an assignment to develop at the workshop. The intended audience for this workshop is computing educators who teach undergraduate computing students at all levels and all courses including those where writing is not typically a course component. No laptop is required for this workshop.

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

15. Computer Science Unplugged

Lynn Lambert, *Christopher Newport University*
Tim Bell, *University of Canterbury*
Thomas Cortina, *Carnegie Mellon University*
Peter Henderson, *Butler University*
Bengt Aspvall, *Blekinge Institute of Technology*

You've been asked to talk to an elementary or high school class about Computer Science, but how can you ensure that the talk is engaging? This workshop introduces Computer Science Unplugged (www.csunplugged.org), a set of kinesthetic, fun activities that cover many core areas of computer science without using technology. Recommended in ACM's A Model Curriculum for K-12 Computer Science, activities in the book include error detection, searching, networks and routing, modularization, and how binary numbers represent characters, FAXes, and images. Participants will see, practice and develop activities from the book and new ones. Participants will also receive the book.

16. Alice 3 and Java for CS1 & AP CS

Wanda Dann, *Carnegie Mellon University*
Stephen Cooper, *Saint Joseph's University*
Don Slater, *Carnegie Mellon University*

This workshop is intended for instructors of CS1 or AP CS courses. This SIGCSE workshop is the first to provide instruction in using Alice3 with built-in integration of Alice and Java. The workshop offers hands-on experience

programming with Alice3 and Java together. Alice3 builds on Alice2's powerful program visualization tool, enabling students to "see" objects and work with object-oriented programming. Participants will learn how to use Alice3 to build virtual worlds and how to use this approach in CS1 courses (introductory programming for majors and non-majors and AP CS). Participants will receive a CD containing Alice3 beta and sample virtual worlds developed as part of NSF-0736552, 0736697, and 0736945 grants. Laptop required, Mac or PC.

17. Roadshow Roadmap: Getting Started with K-12 Outreach

Chris Stephenson, *Computer Science Teachers Association*
Suzanne Menzel, *Indiana University*
Katie Siek, *University of Colorado at Boulder*
Sriram Mohan, *Rose-Hulman Institute of Technology*
Samuel Kamin, *University of Illinois, Urbana-Champaign*

Despite interest in creating and sustaining K-12 outreach programs, many institutions are hampered by a lack of understanding of the K-12 community, resources to develop high-quality presentation materials, and knowledge of current research regarding best practices for encouraging student interest in computing. The primary objective of this workshop is to improve the availability and quality of college outreach programs by providing the attendees with key strategies for creating an effective roadshow. We will share our experiences of working with K-12 teachers and creating viable, self-sustaining outreach programs that connect with K-12 students and have particular relevance for underrepresented groups. Attendees should be interested in developing an outreach program to serve their local area with the aim of addressing the dropping interest in computer science education and careers among young people, and especially among women and minority students. Faculty and staff are encouraged to attend the workshop with students from their school or region who are interested in joining them to create an outreach program. Laptops are strongly encouraged.

18. Mobile Game Development - Engaging Students Early in the Program

Stan Kurkovsky, *Central Connecticut State University*

This workshop presents an approach to teach a semester-long undergraduate course in Mobile Game Development that requires basic Java programming skills as a prerequisite. The goal of this workshop is to introduce faculty to an instructional model that engages students in the course material by using ubiquitous mobile devices relevant to students' everyday lives. By introducing a variety of advanced topics, this course helps convince students that "CS is not just about coding," therefore helping them stay in the major. It is ideally suited for freshmen with AP CS credit for whom it will be their first college-level CS course. Using a mobile phone emulator during the workshop, we will design and develop a simple game, which can then be transferred to and played on a mobile device. The workshop is intended for faculty members teaching introductory undergraduate courses in CS, as well as those looking for ways to improve student retention by introducing them early in a variety of other areas of Computer Science. This workshop will also benefit high school CS teachers looking for an engaging approach to teach the second course in Java. All other faculty members interested in integrating game development as a motivational tool in their curriculum will benefit from this workshop. A Windows laptop with Java 5 or 6 SDK is required.

19. Introducing Bioinformatics Algorithms in Computer Science Courses

Sami Khuri, *San Jose State University*

This workshop is intended for CS educators who would like to get an overview of some of the major techniques used in bioinformatics and a hands-on experience with some of its most popular algorithms. Attendees will be exposed to topics, examples, and problems that can be added to their repertoire of projects for CS1/CS2, data structures and algorithms, and other CS courses, including bioinformatics. The workshop will demonstrate visualization packages for bioinformatics algorithms. The hands-on exercises using online publicly available tools will enhance the understanding of the common bioinformatics tasks. No prior knowledge of biology is needed. Laptop Required.

20. Web Development with Python and Django

Ariel Ortiz, *Tecnológico de Monterrey, Campus Estado de México*

Many instructors have already discovered the joy of teaching programming using Python. Now it's time to take Python to the next level. This workshop will introduce Django, an open source Python web framework that saves you time and makes web development fun. It's aimed at CS instructors who want to teach how to build elegant web applications with minimal fuss. Django is Python's equivalent to the popular Ruby on Rails framework. Topics that will be covered include: setup and configuration, template language, and database integration through object-

relational mapping. Participants should have some familiarity with Python. Laptop required (Windows, Linux or Mac). Further information: <http://django.arielortiz.com/>.

21. MediaScripting: Media Computation In Context

Samuel Rebelsky, *Grinnell College*
Janet Davis, *Grinnell College*
Matthew Kluber, *Grinnell College*

Media computation in CS1 has proven successful in attracting and retaining many kinds of students, in contexts from community colleges to research universities. In this workshop, participants will explore MediaScripting – media computation through scripts (in Scheme or Python) within open-source image editing programs (GIMP and InkScape). Students create and edit images both “by hand” and with scripts, quickly learning that scripting can automate repetitious tasks and provide results that are difficult to achieve otherwise. The MediaScripting approach can fill many roles, from a full CS1 to a week-long module in a more traditional CS1, or an applications or digital art course. Intended audience: We expect this workshop will be appropriate for a wide variety of people, including experienced media computation teachers who want to explore a new approach, college and high school instructors who wish to learn more about the media computing approach, and introductory CS teachers who would like to incorporate a shorter media computing module. Laptop Required: Participants will need a laptop (Linux, Mac/Intel, or Windows) to experiment with scripting. Pairs of participants may also share laptops.

22. Multimedia Projects in Python for Introductory CS Courses

Pamela Cutter, *Kalamazoo College*
Alyce Brady, *Kalamazoo College*

Construction and manipulation of images and sounds is a motivating context for majors and non-majors in introductory computing classes. The algorithms to create interesting multimedia effects provide concrete and interesting results, and easily fit within the constraints of introductory courses’ curricula. The visual and audible feedback provides students an immediate measure of their success, as well as a “cool” factor. This workshop will introduce a series of multimedia examples and assignments that are appropriate for beginning students, using the Python language. Laptop strongly recommended.

23. Computer Security Essentials (Part 2) - Intrusion Detection and System Defense

Paul Wagner, *University of Wisconsin - Eau Claire*
Daren Bauer, *University of Wisconsin - Eau Claire*
Thomas Paine, *University of Wisconsin - Eau Claire*
Jamison Schmidt, *University of Wisconsin - Eau Claire*
Jason Wudi, *JAMF Software*

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 cyberdefense 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. Intended Audience: Computer Science educators interested in developing curricula in the area of computer security and information assurance. This workshop will focus on how computer systems can be hardened against attack, and how system intrusions can be detected. LAPTOP REQUIRED. Each participant's laptop should have wireless network support.

24. Cooperative Learning for CS1 and Beyond: Making It Work for You

Leland Beck, *San Diego State University*
Alexander Chizhik, *San Diego State University*

Would you like to bring the benefits of cooperative learning into your classroom, but have questions about how to do it? If so, this workshop is for you. After a brief overview of cooperative learning principles and research, we will focus on practical issues such as course planning, classroom management, and methods for guiding group processes. Through a series of group exercises, you will be able to experience a cooperative learning process yourself, and also share ideas and experiences with other participants. The principles and activities in the workshop will be applicable throughout the computer science curriculum. In addition, you will receive a complete set of class-tested cooperative learning activities that have raised test scores in CS1 by 25% at our institution. No prior experience with cooperative learning is necessary. Laptop not needed.

25. Intellectual Property Law Basics for Computer Science Instructors

David G. Kay, *University of California, Irvine*

Increasingly the practice of computing involves legal issues. Patenting algorithms, domain name poaching, downloading music, and re-using content from the web all raise questions in the domain of intellectual property (IP) law (which includes patents, copyrights, trade secrets, and trademarks). In the classroom, CS educators often confront questions that have legal ramifications. This workshop will give CS instructors a framework for recognizing the issues, answering students' questions, debunking the most egregious misconceptions about IP, and understanding generally how law and computing interact. We will conclude with an opportunity for participants to share strategies for covering legal issues in the classroom. Laptop not needed.

26. Courseware to Support Teaching a Course in Software Testing

Cem Kaner, *Florida Institute of Technology*
Rebecca L. Fiedler, *Indiana State University*

BBST (Black Box Software Testing) is a mature collection of materials that support online and hybrid courses that we and our colleagues have been teaching in software testing for undergraduates, graduate students and working professionals. We have video lectures, slides, essay and multiple-choice questions, descriptions of in-class activities and an instructor-training program for professional-development instructors that we are adapting for academic use. The materials are available free under a Creative Commons license and freely modifiable. This workshop will introduce you to the collection and to the details of our approach to teaching the BBST courses.

Saturday Workshops, 2:15 p.m. to 5:15 p.m.

27. New Platforms for Teaching Robotics to Computer Scientists

David S. Touretzky, *Carnegie Mellon University*
Ethan J. Tira-Thompson, *Carnegie Mellon University*
Glenn V. Nickens, *Norfolk State University*

This workshop will give participants hands-on experience with two new platforms for teaching the computer science side of robotics to CS undergraduates. The first is an inexpensive hand/eye system easily constructed from a Robotis Bioloid Beginners Kit, a USB interface module, and a webcam. The second is the Chiara, a new educational robot with six legs, a 6-DOF arm with gripper, and an on-board Linux computer; see Chiara-Robot.com for pictures and video. Both platforms are controlled by the Tekkotsu open source application development framework created at Carnegie Mellon and available for free at Tekkotsu.org.

28. Using Mashups to Teach IT Concepts

Mark Frydenberg, *Bentley University*
Suzanne Hansen, *Microsoft*

Mashups—the combining of data from multiple sources into a single application—are popular Web 2.0 constructs, yet are often difficult to create without previous programming skills. But not anymore! This workshop will introduce web-based tools including Microsoft Popfly for building mashups that engage students in hands-on learning of Web 2.0 and information technology concepts. This workshop will also include a discussion of pedagogical aspects of introducing mashups as a teaching tool, and share curriculum resources that the presenter has created for teaching mashups in the introductory information technology classroom. Intended Audience: Faculty teaching courses in (or wanting to learn more about) introduction to technology, Web 2.0, mashups, web-based technologies. Participants should bring their own laptops (PC or MAC). Before attending this workshop, please create a Windows Live ID if you don't have one already.

29. Exploring Wonderland: Teaching with Alice and Media Computation

Barbara Ericson, *Georgia Institute of Technology*
Stephen Cooper, *St. Joseph's University*
Wanda Dann, *Carnegie Mellon University*

This workshop is designed for instructors who are interested in teaching a course combining Alice and Media Computation in Java. The goal of this approach is to provide a complete CS1 that uses Alice to reduce syntax frustration while beginners are learning computing concepts and then uses Media Computation to reinforce the same concepts in Java. We will describe how to assist students in transferring knowledge from Alice to Java. Participants will create some combined Alice and Media Computation projects. For example they will use

chromakey to change the background of an Alice movie. Attendees should be familiar with Alice and Java. Laptop Required!

30. Computer Science Inside... Capturing the Interest of Potential Computer Science Students

Quintin Cutts, *University of Glasgow*
Peter Saffrey, *University of Glasgow*

What fascinates us? That's easy, it's computer science in all its diversity. What fascinates a school pupil? All manner of things, but not necessarily computer science. To get them to engage with our subject, we must find the connections between what interests them and what interests us. This workshop introduces the materials of the Computer Science Inside... project, which has been making these connections and transforming classrooms throughout Scotland. Participants will interact with the project materials and then practice creating engaging connections for themselves, so that they can create the same kind of engagement with computer science whenever they want.

31. Teaching Application Development of Wireless Sensor Networks using Motes and Sensors

Bo Sun, *Lamar University*
Andrew Yang, *University of Houston, Clear Lake*

Integrating practical Wireless Sensor Networks (WSNs) courses into computer science curricula can raise the level of excitement among students. Development of WSN applications, however, is challenging because of interrupt-driven and concurrent nature of WSN applications, severe resource constraints, and no commercial support for TinyOS. This workshop will first give an introduction of WSNs, and then demonstrate how to set up WSN application development environments, run representative WSN applications, and develop meaningful WSN applications. This workshop will be of interest to faculty who are interested in networks and operating systems and try to integrate WSNs into their courses and/or curricula. Participants should bring with them a laptop with pre-installed TinyOS, version 2.1.0 (open-source, freely available from <http://www.tinyos.net>). The tutorial coordinators will provide TelosB motes from CrossBow and WiEye sensors from EasySen.

32. Motivating Computer Science Students with Digital Sound and Music

Jennifer Burg, *Wake Forest University*
Jason Romney, *University of North Carolina, School of the Arts*

This workshop will share MATLAB exercises, Max/MSP exercises, and creative projects that link computer science, music, and practice through digital sound. The creative projects will use Logic (Mac), Cakewalk Music Creator (Windows), and Reason (Mac and Windows). The intended audience is computer science professors who would like to integrate topics related to digital sound into computer science courses. Laptops are not required but can be brought along if the participant chooses. 30-day trial versions of the software can often be downloaded, so participants may want to do this before the workshop. Participants who bring laptops (either Mac or Windows OS) equipped with the software mentioned above can follow along and try exercises that are demonstrated. Participants without laptops or software can still benefit from seeing the demonstrations and going over the worksheets and solutions.

33. Programming Lua

Matthew Burke, *The George Washington University*

The Lua programming language has a (relatively) long history of being used in a variety of programming contexts including game engines, industrial applications and embedded systems. Lua is being used in such high-profile applications as World of Warcraft, Adobe's Photoshop Lightroom, and Nmap. In this workshop you will gain experience both using Lua in a stand-alone manner as well as embedding it in a larger application. In addition, you will experience how Lua can be an effective tool for introducing students to different programming paradigms. The intended audience for this workshop includes the following: 1. Those interested in using an embedded language for scripting another application. 2. Those interested in language design. 3. Those interested in low-resource (embedded) systems. Participants are required to bring a laptop, however, there are no particular software requirements other than a decent text editor (not word processor). Lua source and binaries for Windows, OS X, and various *nixes will be provided.

34. Advanced Game Courses in Computer Science: Getting Beyond Square One with Torque

Brian Ladd, *State University of New York at Potsdam*
Tiffany Barnes, *University of North Carolina at Charlotte*
Dan Cliburn, *University of the Pacific*

Upper-division game development courses are daunting for many faculty: video games are extremely complex and, until recently, game design was not commonly studied in computer science. This workshop presents an overview of the Torque Game Engine, a game-building framework for taming the complexity of game programming, in the context of teaching game design. Attendees will install and use an academically-licensed version of Torque to complete sample assignment as well as examine syllabi from several game development courses; attendees will be prepared to teach an upper-division game development course with a Torque-based project. Recommended workshop prerequisite: Introduction to Game Development. Laptop: Required; MS Windows with 2GB RAM. Intended Audience: People completing Intro Game Courses in CSE; People who have had some experience giving and grading computer game-based assignments and/or computer game-based courses.

35. High-Performance Computing with Microsoft .NET and HPC Server

Joe Hummel, *Lake Forest College*

High-performance computing is becoming increasingly important as academic and commercial institutions replace physical experimentation with computer simulation. In response, universities are starting and growing computational science programs. HPC is a complex field, both in terms of software development and domain modeling. This workshop addresses the former — developing software for HPC systems. In particular, the workshop focuses on the Microsoft tools and technologies available for shared-memory, multi-core programming as well as distributed-memory cluster-based programming. Topics include explicit multi-threading with the .NET Thread class, Parallel Extensions to the .NET Framework (TPL and PLINQ), MPI.NET, and HPC Server 2008. This workshop is intended for newcomers to HPC, and experienced HPC developers looking for an overview of Microsoft's technologies. Laptop not required; attendees wishing to participate will need Windows and any version of Visual Studio 2008 SP1.

36. Automatically Grading Programming Assignments with Web-CAT

Stephen Edwards, *Virginia Polytechnic Institute and State University (Virginia Tech)*
Manuel Pérez-Quñones, *Virginia Polytechnic Institute and State University (Virginia Tech)*

This workshop introduces participants to the use of Web-CAT, an open-source automated grading system. Web-CAT is customizable and extensible, allowing it to support a wide variety of programming languages and assessment strategies. Web-CAT is most well-known as the system that “grades students on how well they test their own code,” with experimental evidence that it offers greater learning benefits than more traditional output-comparison grading. Participants will learn how to prepare reference tests, set up assignments, manage multiple sections, and allow graders to manually grade for design. Go home ready to start using it in your own classes! Prior exposure to a unit-testing framework, such as JUnit, is recommended but not required. Laptop optional.

SIGCSE 2009 Birds of a Feather

The following Birds-of-a-Feather sessions were organized for attendees of the SIGCSE 2009 symposium.

Assessing K-12 and Undergraduate Recruitment Initiatives

David Klappholz, *Stevens Institute of Technology*

Elizabeth Adams, *James Madison University*

Because of the precipitous drop in CS majors, there have been many initiatives aimed at attracting more students, especially young women and members of minority groups. A necessary adjunct to such an initiative is the assessment of its effectiveness. In this BOF we will discuss assessment techniques, both short-and long-term, both shallow and deep, that attendees have used or are planning to use.

CIS Students and Civic Engagement

Elinor Madigan, *Pennsylvania State University*

Jeffrey Stone, *Pennsylvania State University*

The purpose of this BOF is to discuss the use of community-based projects in CIS programs. Our experiences have shown that students working on community-based projects are more engaged in the classroom, exhibit improvements in the quality of their work, and have an increased level of pride in their work. These experiences serve to forge partnerships with community organizations which give rise to internships and job opportunities. These projects have also spurred growth in CIS program interest within non-majors. Students often seek us out to work on these projects, and the local community often looks to us for solutions.

Classroom Assessment Techniques

Edward Gehringer, *North Carolina State University*

Have you ever heard of the “minute paper,” where students spend a minute at the end of class writing about the most important item they learned that day? The minute paper is just one of a set of Classroom Assessment Techniques, which help instructors evaluate how much their students are learning. Other CATs include “muddiest point,” where students are asked which point in the lecture was most confusing, and “pro and con grid,” where they could be asked to list strengths and weaknesses of various sorting methods. Attend this BoF to share your own CATs and learn from your colleagues.

Computational Thinking in the CS Curriculum

Mehran Sahami, *Stanford University*

Owen Astrachan, *Duke University*

Thomas Cortina, *Carnegie Mellon University*

Stuart Reges, *University of Washington*

Computation Thinking (CT) has become a focal point in CS education, both from the perspective of defining the issues we grapple with as computer scientists as well as how these ideas have impacted our curricula. Given the NSF backing for work supportive of CT, many institutions are examining the role it may play in their own curricula. This session provides an opportunity to discuss how CT has impacted changes in CS curriculum and/or coursework. Attendees are encouraged to share their general ideas as well as provide concrete examples of curricular innovation spurred by the notion of CT.

Computer Science: Small Department Initiative

Cathy Bareiss, *Olivet Nazarene University*

Faculty members in a small department (5 FTE or less) face special challenges and other strengths because of their size. This BOF is a time for faculty to talk with other faculty with similar situations and find ways to meet the shared challenges and to take full advantage of their strengths. While the topics may change based on those attending, the first topic of discussion will be ways to try to increase enrollment within the limits of a small department. A second topic might include designing and assessing curriculum with limited numbers of students and faculty.

Connecting Computer Science Education and Music Making

Bill Manaris, *College of Charleston*

The connection between mathematics and music is at least 2500 years old. Computers provide a powerful platform for music analysis and generation. Incoming CS students think of computers more as a music platform, than a number cruncher. Also,

traditional data structures and algorithms may easily be mapped into assignments involving music. This BOF will explore this serendipitous connection. We will discuss existing tools (such as Scratch, ChuckK / miniAudicle, Pure Data, SuperCollider, CSound, jFugue, and jMusic) and explore engaging possibilities for projects, courses, and extra-curricular creative endeavors, such as the Laptop Orchestra.

CS Unplugged, Outreach and CS Kinesthetic Activities

Peter Henderson, *Butler University*

Tim Bell, *University of Canterbury, New Zealand*

Lynn Lambert, *Christopher Newport University*

Outreach activities including Computer Science Unplugged demonstrate computer science concepts at schools and public venues. Computer Science Unplugged is an international project with kinesthetic activities for children, students and adults, using no technology. Concepts include binary number representation, parity, routing and deadlock, public/private key encryption, and others. Effective outreach programs can help counter the idea that computer science = programming or, worse, keyboarding; and can educate the public, interest students, and recruit majors. Come share your outreach ideas that demonstrate how compelling Computer Science is.

Databases for Non-majors: The Next Challenge?

Suzanne Dietrich, *Arizona State University*

Don Goelman, *Villanova University*

With the ubiquity of databases in our information-centric society and the increasing demand for students with interdisciplinary skills, the next challenge for database educators may be designing courses for non-majors. In this session, the leaders will moderate a lively discussion on goals, topics, successes, and disappointments. The challenge in non-major offerings is to enable a diverse audience of students with the knowledge of databases and information exchange, while engaging future chemists, musicians, statisticians, and criminologists.

Distance Education: Translating the Classroom Experience to Online Delivery

William Hochstetler III, *Franklin University*

Daniel Bell, *Franklin University*

A significant challenge in distance education is translating the classroom experience to an online environment. This session will explore techniques that have been used successfully to deal with representative issues such as student to student and student to professor interaction, sensing non-verbal communication, office hours and supplemental material such as multimedia. The value of using pre-recorded or live lecture sessions and the larger issue of synchronous vs. asynchronous activities will be explored, including the use of innovative technology, such as Internet based collaboration tools. Both the experienced or merely curious in engaging students online are encouraged to attend.

Education, Computers and Society

Florence Appel, *Saint Xavier University*

Joseph Oldham, *Centre College*

This session is organized by ACM SIGCAS (Computers and Society) for those interested in addressing the social and ethical consequences of computing within their curricula. We will emphasize sharing resources and ideas, and strengthening our network. Discussion items will include best practices and approaches to teaching computer ethics and social impact, developing programs for recruitment and retention of under-represented populations, and implementing service learning course components. We will also address ways in which SIGCAS and SIGCSE can collaborate to support each other's work, including organizing a future pre-SIGCSE conference dedicated to computers and society issues.

Flattening Computer Science: Internationalization of CS Curriculum

Jens Mache, *Lewis & Clark College*

Virginia Lo, *University of Oregon*

Cynthia A. Brown, *Portland State University*

Steve Tanimoto, *University of Washington*

Jane Chu Prey, *Microsoft Research*

J. Kent Foster, *Microsoft*

Amanda Camp, *Google*

Should CS graduates be fluent in Java and Chinese? How do we teach cross-cultural communication, international computer ethics, and universal HCI? How can we best provide globally distributed software development experiences? Come

brainstorm these critical questions with us. This discussion started as part of the CPATHi18n.org project sponsored by NSF. Our project has the goal of transforming CS education through internationalization (or i18n, borrowing the abbreviation from the software development community). We are building a Pacific Rim community to implement these ideas. We want to hear your experiences, your creative thoughts, and your vision of the model internationalized CS curriculum.

The Future of the Advanced Placement Computer Science Program

Laurie White, *Mercer University*

Paul Tymann, *Rochester Institute of Technology*

Barbara Ericson, *Georgia Institute of Technology*

With the discontinuation of the Advanced Placement Computer Science (APCS) AB examination after 2009, there has been much discussion about the future of the APCS program. Members of the APCS Test Development Committee will lead a discussion of the future of the program in the next five years until the results of the Course and Exam Review Commission can be implemented.

High School, post AP C.S. AB: Data Structures

Lon Levy, *Oregon High School*

Todd O'Bryan, *duPont Manual High School*

The College Board has decided to make the May 2009 AP Computer Science AB: Data Structures Exam the last one. While the AP C.S. A course will continue, those schools that offer data structures must modify their curricular offerings. Some high schools may be dropping computer science completely. Other schools are attempting to articulate a course with regional universities. This BOFS is an opportunity for high school data structures teachers and the professors who support such programs to come together and discuss the different approaches we are considering.

How to add a computer science requirement to our high schools

Baker Franke, *University of Chicago Laboratory Schools*

Matt Brenner, *Sidwell Friends School*

Jeff Solin, *Northside College Preparatory High School*

Marty Billingsley, *University of Chicago Laboratory School*

Don Yanek, *Northside College Preparatory High School*

Vincent Nguyen, *Northside College Preparatory High School*

This session is for anyone who is interested in seeing computer science become an academic requirement in high school. There are many hurdles to leap and a range of questions to answer, from fundamental questions like “what should be taught?” to practical questions like “where does it fit into a student’s schedule?” Our discussion leaders represent groups that are at various stages in the process. Come help those who are trying to build a CS requirement, brainstorm with others in your position, or come hear about the issues that you may encounter if you’re thinking about proposing a requirement.

Human Communication in Computer Science & Software Engineering Education

Clifton Kussmaul, *Muhlenberg College*

Paul Anderson, *Miami University (Ohio)*

Janet Burge, *Miami University (Ohio)*

Communication – between humans – remains at the heart of software development, yet most CS and SE courses and programs focus almost exclusively on theory, tools, and processes. This BOF will enable colleagues from various institutions to share concerns, ideas and best practices for providing students with the communication skills needed in the profession. First, we will summarize a 2008 NSF-sponsored Chautauqua on communication in software engineering curricula (10 min). Next, we will form subgroups to explore specific issues and opportunities (20 min). We will conclude by discussing the results of the subgroups (10 min).

Increasing Enrollment in High School Computer Science Courses

Michael Scott, *University of Texas at Austin*

Stacey Armstrong, *Cypress Woods High School*

High school computer science teachers face many challenges. The subject matter they teach is often misunderstood by counselors and other teachers. They are often the only computer science teacher at their school. The department they belong to varies from school to school. The goal of this BOF session is to share successful practices high school teachers have used to increase enrollment in their courses and programs.

iPhone Application Development

Michael Rogers, *Millikin University*

The beauty of Apple's iPhone is more than skin-deep: developing for the iPhone is a great opportunity for Computer Science students to see a world-class framework in action, and then get world-wide recognition by selling their work on the App Store. This Birds of a Feather session will bring together those faculty who are teaching, or (more likely, considering the newness of the product), contemplating teaching, iPhone Application Development.

The Last Lecture: A Legacy's Impact on Recruiting, Retention, and Learning

Carol Paczolt, *Purdue University*

Mindy Hart, *Purdue University*

The Computer Science world is certainly already familiar with the story of Randy Pausch. The familiarity is not only because of his fight with pancreatic cancer and best-selling book, *The Last Lecture*, but also because of his contributions to the field of computer science through programs such as ALICE. This BOF will provide an opportunity to discuss the potential impact his story can have, especially as a recruiting or teaching tool.

Mathematical Reasoning in Computer Science

Peter Henderson, *Butler University*

Doug Baldwin, *SUNY Geneseo*

Murali Sitaraman, *Clemson University*

Mathematics and mathematical reasoning are central to computer science, and, indeed, are becoming increasingly prominent in computing curriculum guidelines. This birds-of-a-feather is a forum for educators to discuss the role of mathematics in computer science, information technology, and software engineering curricula. Discussion topics may include: new curriculum models, new courses, mathematics early, the role of mathematics both in particular courses and throughout the curriculum, available resources, and promoting future change. Opposing as well as supporting views are welcome.

Multi-Core Programming in the Curriculum

Barry Wilkinson, *The University of North Carolina at Charlotte*

Barry Kurtz, *Appalachian State University*

Clayton Ferner, *University of North Carolina, Wilmington*

Jens Mache, *Lewis & Clark College*

Rahman Tashakkori, *Appalachian State University*

Yaohang Li, *North Carolina A&T State University*

For the last few years, computers have been manufactured with multiple cores. This trend is likely to continue with increasing numbers of processor cores, now that the practical limits of processor clock frequency has been reached. Hence, the present and next generation of computer science students must be prepared for this computing platform. In this BoF, participants will exchange ideas on how best incorporate programming of multicore computer systems into the curriculum. Should it be introduced into CS 1/2 or, as is usual for new techniques, introduced as a separate upper-level course, and what should be introduced?

Organizing the Algorithm Visualization Community

Cliff Shaffer, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Tom Naps, *University of Wisconsin, Oshkosh*

Stephen Edwards, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Algorithm visualizations (AVs) have long held great promise for improving CS Education. Following on our popular BoF session from SIGCSE'08, this year's goal is a focused discussion on how to engage the AV community in an organized effort to improve AV development and use. We will briefly describe our NSF-sponsored project to build the AV community, and present proposals from our Steering Committee Meeting. Primarily we will solicit community feedback on how to increase involvement in cataloging available AVs, defining AV best practice, and disseminating the Open AlgoViz sourceforge project. See the AlgoViz Wiki (<http://algoviz.cs.vt.edu>).

Past, Present, and Future Trends in Gaming and Computing Curricula

Jon Preston, *Southern Polytechnic State University*

Briana Morrison, *Southern Polytechnic State University*

Michael McCoy, *The Guildhall at Southern Methodist University*

Digital media and game design/development courses, and game degree programs are on the rise. Who is involved, and what is taking place in the field of gaming and computing education? Some institutions use game-related content to improve enrollment and retention; some use gaming in introductory courses or as upper-division specializations; others offer complete degrees. Organizations such as the IGDA's SIG on Education are working to support and provide guidance, and it would serve SIGCSE well to be involved. This discussion group covers current trends/activities, future plans, and lessons learned from existing programs.

Putting the Fun Back into CS1

Paige Meeker, *Presbyterian College*

Many students with an initial interest in a Computer Science major often leave the major after taking the introductory course. These students complain that the material is difficult and boring. This BOF will provide a discussion of how CS professors can create a more enjoyable learning experience for introductory students. I hope that the discussion will provide the participants with ideas for new assignments, new topics, or new techniques for providing an interactive and interesting CS1 experience.

Python in Education

Michael H. Goldwasser, *Saint Louis University*

Claude Anderson, *Rose-Hulman Institute of Technology*

Curtis Clifton, *Rose-Hulman Institute of Technology*

Richard Enbody, *Michigan State University*

Barbara Ericson, *Georgia Institute of Technology*

Mark Goadrich, *Centenary College of Louisiana*

Andrew Harrington, *Loyola University Chicago*

Bradley N. Miller, *Luther College*

David L. Ranum, *Luther College*

Christine A. Shannon, *Centre College*

The use of Python has been growing in both industry and computer science education. Its popularity stems from its combination of an elegantly simple yet consistent syntax, together with a powerful and robust core. Python's has been adopted at various institution types (high school, small college, large university), to support various courses (CS0, CS1, CS2, advanced topics), programming paradigms (procedural, object-oriented, functional) and application areas (multimedia, robotics, scientific computing). This BOF will allow educators to discuss their experiences using Python in the classroom, and its potential integration along side other languages in the CS curriculum.

Roadshow Roadmap: Guide for Established Outreach Programs

Suzanne Menzel, *Indiana University*

Bill Hogan, *Cornell University*

Katie Siek, *University of Colorado at Boulder*

Many institutions have initiated K-12 outreach roadshow programs, following the lead of Carnegie Mellon University. As the programs mature, issues arise in how to evolve the content, obtain funding and institutional support, reward the participants, collect and organize data, and assess the program. The primary objective of this discussion group is to provide insights, advice, and encouragement from people experienced in dealing with all these issues. We also wish to learn what other people are doing at other schools so we can all benefit from the collected wisdom.

Scratch in the Undergraduate CS Curriculum

Ursula Wolz, *The College of New Jersey*

John Maloney, *Massachusetts Institute of Technology*

Scratch (<http://scratch.mit.edu/>) is a programming environment that combines “drag and drop” program construction with a web-based social network. Designed to develop 21st century learning skills, it allows creators of all ages to construct two-dimensional stories, games, animations, music and art. Scratch is used formally in classrooms around the world, but also supports millions of informal programmers via its on-line community. At SIGCSE-08 a standing-room-only BOF discussed what place Scratch might have in the introductory undergraduate curriculum. This year we will address how the Scratch community can support engagement and broaden participation in computing in both formal and informal settings.

SIGCSE at 40: The Past and the Future

John Impagliazzo, *Qatar University*

Robert Aiken, *Temple University*

Nell Dale, *University of Texas at Austin*

This SIGCSE Symposium marks the Fortieth Anniversary of the birth of SIGCSE. This session will host an interactive discussion among the SIGCSE's attendees and its veterans to exchange ideas on ways to advance the tenets of the organization in light of its historical beginnings. A significant number of SIGCSE pioneers should be present to interact with the attendees. The expected spirited interaction will allow a sharing of ideas to help SIGCSE in the development of its future.

Sustaining Efforts to Broaden Participation in Computing

Tiffany Barnes, *University of North Carolina at Charlotte*

Teresa Dahlberg, *University of North Carolina at Charlotte*

Research shows that efforts that engage students in outreach, research, and service are effective for student retention and have positive effects on recruiting. However, such efforts are not part of the typical computer science curriculum and are often discontinued after grant funding runs out or the faculty champion leaves. We believe that a fundamental shift is needed in computing culture, and it can be achieved by building a community that engages students and professionals in using computing to benefit humanity. This BOF will discuss ways to build and sustain this and other effective practices to broaden participation in computing (BPC).

Teaching Globally Distributed Collaboration in a Classroom Setting

Gary Thompson, *San Francisco State University*

Software engineering professionals are increasingly required to work and collaborate with colleagues on teams that are dispersed both nationally and globally. This BOF will present an opportunity for CS instructors to share, discuss and explore pedagogical techniques designed to introduce students to and educate them in the real world issues of working in groups that span multiple time zones, languages and cultures. The discussion will include real examples of globally distributed student development teams as well as techniques developed to assess the level of teamwork within software development teams, both distributed and local.

Teaching Track Faculty in CS

Donald Slater, *Carnegie Mellon University*

Dan 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 their experiences as members of the faculty of their departments and schools, and to provide opportunities for schools considering such positions to gather information.

Teaching Web Programming

Marty Stepp, *University of Washington*

Jessica Miller, *University of Washington*

Last year we had an informative discussion about web programming and its place in the computer science curriculum. Join us for a continuation of that conversation and discussion of new topics from the past year. Possible topics for discussion include: how web programming should be incorporated into the computer science curriculum; whether it should be required; which languages and technologies should be covered; what materials should be used; and challenges unique to teaching web programming. Also up for discussion are relevant web events and technologies from the past year, such as Chrome, iPhone, Android, and ECMAScript 3.5.

Teaching with Alice

Donald Slater, *Carnegie Mellon University*

Wanda Dunn, *Carnegie Mellon University*

Stephen Cooper, *National Science Foundation*

This session is for anyone currently using Alice, or exploring the possibility of using Alice in their curriculum. The discussion leaders and experienced Alice instructors will share teaching strategies, tips, and tricks with each other and those

new to Alice. The session provides an arena for sharing Alice instructional materials and ideas for courses at all educational levels. This is an opportunity to share assignments and pointers to web sites where collections of instructional materials, such as syllabi, student projects, exams, and other resources are available. In addition, Alice 3 (the latest version) will be a topic of discussion.

TeachScheme/ReachJava

Viera K. Proulx, *Northeastern University*

TeachScheme/ReachJava is an unconventional introductory computing curriculum that introduces both functional programming and the object-oriented program design in a systematic design-driven style. The design recipes for designing data, designing functions/methods, and designing abstractions empower both the student and the instructor. The ReachJava part focuses on understanding data design and adds Java language features only when needed. Instructors experienced in using TeachScheme/ReachJava curriculum will be invited to share their experiences and course materials with each other. Those interested in learning about TeachScheme/ReachJava will have the opportunity to ask questions, see examples of student work, and learn more about the TeachScheme/ReachJava pedagogy.

Technology that Educators of Computing Hail (TECH)

Dan Garcia, *University of California, Berkeley*

Don Bailes, *East Tennessee State University*

Sally Fincher, *University of Kent*

The judicious use of technology in computing education (in and out of the classroom) can be empowering and transformative. However, it is very difficult to discover what tools are available and how effective they have been. The ACM Education Council hopes to address this with a new website, “Technology that Educators of Computing Hail (TECH)”, that will serve to provide a central, organized collection of links to teaching resources. It will feature search, rating, tagging and commentary. The goal of this BOF is to collect “experience reports” from educators who have used technology for teaching computing – with success or not!

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

Gloria Childress Townsend, *DePauw University*

Paula Gabbert, *Furman University*

In January 2004, we organized the second SIGCSE Committee (“Expanding the Women-in-Computing Community”). Our annual Town Meeting provides dissemination of information concerning successful gender issues projects, along with group discussion and brainstorming, in order to create committee goals for the coming year. We select projects to highlight through listserv communication and through our connections with NCWIT, ABI, ACM-W, etc. This year we anticipate project information from the Coalition to Diversify Computing and from the Empowering Leadership Alliance and other major organizations dedicated to expanding the community of women and minorities in computing.

Web-CAT User Group

Stephen Edwards, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Manuel Pérez-Quñones, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Web-CAT is the most widely used open-source automated grading system and winner of the 2006 Premier Award, recognizing high-quality, non-commercial courseware for engineering education. Web-CAT is customizable and extensible. It supports a wide variety of programming languages and assessment strategies. Web-CAT is famous for “grading students on how well they test their own code,” but it can do much more. This BOF will allow existing users, new adopters, and those trying to choose an automated grader to meet, share experiences, and talk about what works and what doesn’t. Information on getting started quickly with Web-CAT will also be provided.

SIGCSE 2009 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 6, 10:00 – 12:00, Exhibit Hall B

A PCIL to Assist in Algorithm Visualization

Martha Kosa, *Tennessee Technological University*
Brandon Malone, *Mississippi State University*
Frank Hadlock, *Tennessee Technological University*

Broadening Participation in Computing: Increasing the Retention of Minorities and Women in First Year CS Courses

Dr. Aurelia T. Williams, *Norfolk State University*
Dr. Sandra J. DeLoatch, *Norfolk State University*

CAFE: A Collaboration and Facilitation Environment for Engaging Students in Computer Science

Sarah Monisha Pulimood, *The College of New Jersey*
Ursula Wolz, *The College of New Jersey*
Kim Pearson, *The College of New Jersey*
Andrew Chiusano, *The College of New Jersey*

Closing the Feedback Loop: Using Historical Data and Student Feedback to Guide Course Changes

Adrienne Decker, *University at Buffalo, SUNY*
Carl Alphonse, *University at Buffalo, SUNY*

Computer Science Illustrated

Ketrina Yim, *University of California, Berkeley*
Dan Garcia, *University of California, Berkeley*
Sally Ahn, *University of California, Berkeley*

CS Assessment: Goals, Learning Outcomes and Rubrics

Frances Bailie, *Iona College*
Adel Abunawass, *University of West Georgia*
Bill Marion, *Valparaiso University*
Caroline St. Clair, *North Central College*
Deborah Whitfield, *Slippery Rock University*

Current State of K-12 Computer Science Outreach Programs

Elizabeth Adams, *James Madison University*

Debugging in Scratch: Thinking about Programming Rather than Thinking about Syntax

Colleen Lewis, *University of California, Berkeley*

Design and Implementation of a Studio-Based Learning Model for CS2

Dean Hendrix, *Auburn University*
Lakshman Myneni, *Auburn University*
Hari Narayanan, *Auburn University*
Margaret Ross, *Auburn University*

Ensemble: Creating a National Digital Library for Computing Education

Peter Brusilovsky, *University of Pittsburgh*
Lillian Cassel, *Villanova University*
Steve Carpenter, *Texas A&M University*
Lois Delcambre, *Portland State University*
Steve Edwards, *Virginia Polytechnic Institute and State University (Virginia Tech)*
Patrick Fan, *Virginia Polytechnic Institute and State University (Virginia Tech)*
Ed Fox, *Virginia Polytechnic Institute and State University (Virginia Tech)*
Rick Furuta, *Texas A&M University*
Dan Garcia, *University of California, Berkeley*
Greg Hislop, *Drexel University*
Maggie Johnson, *Google Corporation*
Joe Lucia, *Villanova University*
David Maier, *Portland State University*
Manuel Pérez-Quñones, *Virginia Polytechnic Institute and State University (Virginia Tech)*
Steve Seidman, *University of Central Arkansas*
Frank Shipman, *Texas A&M University*
Chris Stephenson, *Computer Science Teachers Association*
Heikki Topi, *Bentley College*
Carol Weiss, *Villanova University*
Bryant York, *Portland State University*

Examining the Behavior of Effective Student Programmers

Jason Snyder, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Exploring CPU Design with DLSim Plug-ins

John Donaldson, *Oberlin College*
Richard Salter, *Oberlin College*
Joseph Kramer-Miller, *Oberlin College*
Serguei Egorov, *Oberlin College*
Akshat Singhal, *Oberlin College*

Exposing the Data in an Automated Grading System Using Open Source Reporting Tools

Anthony Allevato, *Virginia Polytechnic Institute and State University*
Stephen Edwards, *Virginia Polytechnic Institute and State University*

From Class Project to Open Source: Packaging for Outside Development

Edward Gehringer, *North Carolina State University*
Adam Budlong, *North Carolina State University*
Ashwin Veeravalli, *North Carolina State University*

Integrated Courses in Dalhousie University's Bachelor of Informatics

Ernst Grundke, *Dalhousie University*
Christian Blouin, *Dalhousie University*
Michael McAllister, *Dalhousie University*
Norman Scrimger, *Dalhousie University*
Jacob Slonim, *Dalhousie University*

Internationalization of Computer Science Education: The Pacific Rim Community Model

Virginia Lo, *University of Oregon*
Cynthia Brown, *Portland State University*
Steve Tanimoto, *University of Washington*
Jens Mache, *Lewis and Clark College*
J. Kent Foster, *Microsoft*
Jane Chu Prey, *Microsoft Research*
Larry Snyder, *University of Washington*

Just the Necessary Facts: Algorithm Visualization for Artificial Intelligence

Laurie White, *Mercer University*

Legacy vs. Cutting Edge Technology in Capstone Projects: What Does Work Better?

Carsten Kleiner, *University of Applied Sciences & Arts Hannover*

Arne Koschel, *University of Applied Sciences & Arts Hannover*

Leveraging Pods as a Pedagogy Tool to Facilitate Multicultural Collaborative Undergraduate Research in Multi-University Partnerships

Jeremy Barksdale, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Gerry Dozier, *North Carolina A&T State University*

Kelvin Bryant, *North Carolina A&T State University*

Scott McCrickard, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Maureen Biggers, *Indiana University*

Woodrow Winchester, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Living In the KnowlEdge Society (LIKES)

Edward Carr, *North Carolina A&T State University*

Edward Fox, *Virginia Polytechnic Institute and State University (Virginia Tech)*

Robert Beck, *Villanova University*

Teaching Web-based Information Retrieval/Filtering Course to CS Undergraduates: Projects from Collaborative Filtering Based Recommenders

Seikyung Jung, *Westfield State College*

Understanding NXT Mindstorms Robots For Use in Upper-level CS Courses

Richard James, *Rollins College*

Nickolas Nelson, *Rollins College*

Video Scenarios: Developing a Connection between Computing and Creative, Real World Problem Solving

Madalene Spezialetti, *Trinity College*

Session II: Friday, March 6, 2:45 – 4:45, Exhibit Hall B

Action Labs for Computer Science: A Look Inside

Crystal Edge, *Coastal Carolina University*

Stephen Sheel, *Coastal Carolina University*

An Experience Report on Teaching Embedded System for CS and SE

Kai Qian, *Southern Polytechnic State University*

Andy Wang, *Southern Polytechnic State University*

Business Literacy for Computer Science Students: Bridging the Business/Technology Gap

Tony Pittarese, *East Tennessee State University*

Children and Software: Mediating Computing Concepts for Engagement and Learning

Giovanni M. Bianco, *University of Verona, Italy*

Tim Bell, *University of Canterbury, New Zealand*

Simonetta Tinazzi, *Secondary School L. da Vinci*

Children's Design Ideas for a Novice Programming Tool

Sureyya Tarkan, *University of Maryland*

Vibha Sazawal, *University of Maryland*

Collaborative Software Development for Novel Devices

Roy Pargas, *Clemson University*

Kyungsoo Im, *Clemson University*

Samuel Bryfczynski, *Clemson University*

CS 1 with Graphics from Day One: A Typical Graphics Library, In-Class Examples and Assignments

Michael Main, *University of Colorado in Boulder*

Design of a Facility for Sharing Resources for Educational Assessment in Computer Science

Steven Tanimoto, *University of Washington*
Jamie Cromack, *Microsoft Research*
Stephen Kerr, *University of Washington*
Hiroki Oura, *University of Washington*
Tyler Robison, *University of Washington*

Engaging Computer Science Students through Cooperative Education

James Huggins, *Kettering University*

GraphPad: A Graph Creation Tool to Expose Student's Cognitive Processes in CS2/CS7

Samuel Bryfczynski, *Clemson University*
Roy Pargas, *Clemson University*

Introducing High School Chemistry Students to Computer Science

Christina Williams, *Colorado State University*
Michelle Strout, *Colorado State University*

iProjectBuddy: Mobile Device-Based Communication Tool for Effective Team Collaboration

Kyungsoo Im, *Clemson University*
Tacksoo Im, *Clemson University*

JavaWIDE - The Java Wiki Integrated Development Environment

Jam Jenkins, *Georgia Gwinnett College*
Sonal Dekhane, *Georgia Gwinnett College*
Nannette Napier, *Georgia Gwinnett College*
Jim Rowan, *Georgia Gwinnett College*
Brian Ladd, *State University of New York at Potsdam*

Kinesthetic Computer Science Activities in a Virtual World

Tim Bell, *University of Canterbury, New Zealand*
Daniela Marghitu, *Auburn University*
Mick Grimley, *University of Canterbury, New Zealand*
Hiroki Manabe, *Kanagawa Vocational Training School for Persons with Disabilities*
Giovanni Bianco, *University of Verona, Italy*

PALMS: Problem-Oriented Animated Learning Modules for CS1

Jeffrey Stone, *Pennsylvania State University*
Tricia Clark, *Pennsylvania State University*

Q6C: A Multidisciplinary Approach for Teaching Online Research Practices

Katherine Deibel, *University of Washington*
Sarah Read, *University of Washington*
Tim Wright, *University of Washington*

Qatar Summer College Preview Program: An Information Systems Project Perspective

Raja Sooriamurthi, *Carnegie Mellon University*
Gloria Hill, *Carnegie Mellon University*

Research Mini-Projects for High School Students

Tzu-Yi Chen, *Pomona College*

System Development: A Project Based Approach

Hossein Hakimzadeh, *Indiana University, South Bend*
Robert Batzinger, *Indiana University, South Bend*
Susan Gordon, *Indiana University, South Bend*

Teaching Operating Systems/System Administration Using Windows PowerShell

E. Kent Palmer, *Wingate University*

Toward a Threads-Based Computing Curriculum

Andy Wang, *Southern Polytechnic State University*

Kai Qian, *Southern Polytechnic State University*

WAPPEN: A Web-Based Application Framework for Learning Programming

Koji Kagawa, *Kagawa University*

Wireless Tablet PC-Based Enhancement to Student Learning in Electrical Engineering, Computer Engineering, and Computer Science Courses

Weiyang Zhu, *Hampton University*

Chutima Boonthum, *Hampton University*

Jean Muhammad, *Hampton University*

Robert Willis, *Hampton University*

SIGCSE 2009 SRC Graduate Student Research Abstracts

High-level Robot Arm Programming Primitives

Glenn V. Nickens, *Norfolk State University*

Access to the field of robotics has been limited in computer science undergraduate programs due to the cost of robots, and complexity of the underlying mathematics and theory that is needed to implement kinematics, path planning and grasping objects. Researchers in the field of robotics are addressing this accessibility problem through the development of software frameworks, like Tekkotsu, that provide a level of abstraction. In Tekkotsu, programmers use high-level primitives, such as "nod your head" and "walk to that location", to control robots. This research is focused on developing high-level manipulation primitives for a 3-link planar arm.

Evaluating Games for Teaching Computer Science

Michael John Eagle, *University of North Carolina at Charlotte*

There has been a sharp decline in computing enrollments since 2000. Many educators have started to look to video games to help motivate students. However, the study of games has lacked a coherent research paradigm. In this paper, we use methods from the Intelligent Tutoring System literature to evaluate an educational computer science game. This is done by mapping empirical learning curves on student game-log data collected in a follow up study on the game *Wu's Castle*. We believe the results of this type of analysis can help evaluate and improve educational video games.

Attracting African American Students into the Computer Science Pipeline

Tonya R. Groover, *University of Pittsburgh*

Computer science (CS) education is in a state of turmoil. The number of students entering undergraduate degree programs is shrinking. There are dramatic disparities in access and exposure to computing. Secondary CS programs do not reflect the breadth and depth of the field. This research addresses these problems by enhancing the quality and quantity of urban, African American students entering into the computer science pipeline (i.e. the curriculum and opportunities available to engage and learn computer science). Using the recommendations and framework from the ACM Model Curriculum for K-12 Computer Science, we designed and implemented curricula modules to address these problems.

Abstraction and Reduction Techniques for Compositional Verification

Haiqiong Yao, *University of South Florida*

Compositional verification is essential for verifying large systems. However, approximate environments are needed when verifying the constituent components in a system. Effective compositional verification requires finding a simple but accurate over-approximate environment for each component. This proposal exploits several techniques of abstraction and symmetry to address the state explosion problem in concurrent systems with interchangeable and unordered components. The advantage of these techniques is not to introduce extra behavior involved in spurious counterexamples, which contributes less computation to confirm the found counterexamples. The experiments on several large asynchronous circuits show that these techniques effectively reduce state space of each component and demand on less resources.

Fair Electronic Exchange based on Biometrics

Harkeerat Bedi, *University of Tennessee at Chattanooga*

Fair exchange between two parties can be defined as where: either both parties acquire what they expected or neither does. Protocols that facilitate such transactions are known as "fair exchange protocols". Our proposed protocol extends such an exchange protocol by providing client authentication using a finger print based

authentication technique and uses symmetric key algorithms for encrypting messages instead of asymmetric ones thus improving efficiency. Furthermore, our protocol also uses an invisible trusted third party (TP) that intervenes only when disputes in a transaction occur. This significantly reduces overhead as the TP stays dormant most of the time.

Reputation Based Routine in Wireless Ad Hoc Networks

Alma Cemerlic, *University of Tennessee at Chattanooga*

We develop a fine grained reputation management system for wireless ad hoc networks in order to improve connectivity and cooperation among the nodes and to minimize interaction with malicious nodes if they are present in the network. Reputation mechanism is based on constant monitoring and exchange of reputation information. A combination of observations and reports is used for making routing decisions. We use the Dirichlet distribution with Bayes theorem to provide a finer granularity for classification of nodes based on their behavior. We compare our approach with the simpler reputation implementations which merge all types of misbehaving nodes into one category.

SIGCSE 2009 SRC Undergraduate Student Research Abstracts

Using Verlet Integration for Physics Engine Development

Michael Gibeau, *Central Connecticut State University*

This project focuses on designing and implementing a mathematically simple yet highly interactive framework for simulating physical objects in two-dimensional space. Tools allow for the creation of objects and the world in which they exist and interact with. Verlet integration is used to simulate the movement of complex objects constructed of many elementary components. A single object collision algorithm is used to simulate any of the objects that are designed. To test the viability of the framework, a computer game has been designed which shows the level of object interactivity with the created world.

Investigating Automated Exploration and Searching

Andre Mitchell, *Coe College*

Exploring and searching for targets within an environment is a key component when developing sophisticated artificial intelligence. If we can understand in detail how our brain performs certain operations, such as observing the environment, analyzing and collecting clues about where things can be found within the environment and searching for particular targets, we will be able to create automated entities which explore environments similarly to the way that we humans do. This allows us to create powerful simulators which can be used to test the efficiency of the design of a large construction before it is built, like a shopping mall. By modifying and customizing the A-start shortest path algorithm I have developed an algorithm that takes into consideration the exploration factor. Entities driven by physiological need explore an unknown environment in search for targets to satisfy those needs.

Global Illumination with Slow Light Transport

David Schroeder, *Coe College*

Silmaril is a rendering engine capable of simulating various forms of indirect light transport while allowing the photons to travel at a finite speed, causing various lighting effects to emerge which are imperceptible in the real world, and which may be helpful in the explanation of relativity and the speed of light. Path tracing starting at light sources is used to ensure the proper appearance of illumination effects. While en route to creating Silmaril, I created both a traditional reverse raytracer, as well as a more advanced path tracing engine that supports sophisticated global illumination.

Performance of General Graph Isomorphism Algorithms

Sara Voss, *Coe College*

Graphs are a powerful tool in pattern matching. A significant number of algorithms are presented in literature, but few papers characterize their performance. Consequently, it is not known how the algorithms are affected by the type, size, and node numbering variations presented in real applications. This abstract presents a benchmarking activity for five exact, general graph isomorphism algorithms which aim to be practical solutions. The graphs considered are highly structured and sizes range from a few nodes up to 16,000 nodes. The affect of the numbering of the nodes in a graph is also explored.

COMTOR: Comment Mentoring System – Evolving a Computer Science Educational Tool

Autumn Breese, *The College of New Jersey*

COMTOR, Comment Mentoring System, is a prototype computer science educational tool being developed at The College of New Jersey. COMTOR is comprised of a number of test modules that can be run on software packages to give feedback on the quality of the documentation and code. The project addresses two main problems: developing a set of metrics to judge student code to allow COMTOR to provide meaningful feedback,

and evolving the system by adding or adjusting features that will meet the goal of assisting students learning to program. The end goal of the project is to develop COMTOR into a powerful computer science educational tool that is ready for deployment to students on a larger scale.

An Online Collaboration Environment for Interactive Journalism

Andrew T. Chiusano, *The College of New Jersey*

This project entails the design and implementation of an online collaboration environment, CAFE, specifically designed for students participating in a pilot program designed to increase students' interest in the computing sciences using interactive journalism. CAFE has a unique workflow system that allows users to work together and publish an online magazine. Security and privacy were high priorities while developing the system, which is a web application built using PHP and the DBMS PostgreSQL. Currently, CAFE is being adapted for use in college classrooms.

Evolving the User Interface of a Computer Science Educational Tool

Ruth Dannenfelser, *The College of New Jersey*

COMTOR, comment mentor, is a web-based system that generates a comment usage report based on a user's submission of Java source code. The user interface of this system needs to be designed for the main two users of the system, students and teachers of introductory computer science courses. Issues that need to be addressed in the design of this interface are universal accessibility and the potential for expansion as new modules are added. Usability testing and interviews will determine the weaknesses of the current interface and help in the development of the new interface.

Comparing Numerical Integration Methods in a Simulator for the Draping Behavior of Cloth

Michael Liberatore, *The College of Wooster*

The soft-body dynamics of cloth, water, and smoke are often implemented using interconnected particles to which forces can be applied. In such simulations, the particle positions are often approximated using the fourth order Runge-Kutta (RK4) integrator. However, some modern simulations rely on the less complex and non-physically based Verlet integration technique. We compare the efficiency, computational speed, and impact on animation frame rates in cloth models to determine the benefits of using Verlet versus an RK4 integrator. We analyze the folding behavior of cloth for both integrators using a robust user controlled simulation environment that we created.

Managing Graphics Processing Units in High Throughput Computing Environments by Integrating Graphics Processing Unit Discovery with Condor

Timothy Blattner, *Marquette University*

In a default installation of Condor, machine configurations do not send information regarding GPUs connected to a HTC environment. GPUs are powerful tools and would make an effective addition to a HTC environment. To add necessary parameters to include GPU statistics users must manually configure Condor. In order to integrate GPU discovery into Condor, a machine will provide information regarding the GPUs connected to the system within the ClassAds sent to the Condor collector. From the details provided, the ClassAds will be modified and sent to a pool's collector so jobs accept parameters involving various GPU specifications.

Real-Time TCP for Embedded Devices

Aaron Gember, *Marquette University*

An increase in both multimedia based network services and embedded devices is presenting new challenges for existing network protocols. While the widely used Transmission Control Protocol (TCP) provides the data delivery guarantee necessary for streaming media, it lacks the necessary timeliness. The development of a real-time TCP for embedded devices satisfies soft real-time constraints. Designed for Xinu - an educational

embedded operating system - real-time TCP extends the existing TCP protocol in a straightforward style, mindful of embedded system constraints.

XML Based Grammar and Language Conversion

John-Paul Verkamp, *Rose-Hulman Institute of Technology*

This project aims to leverage existing techniques such as the extensibility of XML and XSL Translations to build an easily extensible conversion tool capable of translating between SQL and Relational Algebra. The primary goal in doing so is to expose the transitional stages so as to educate introductory databases students into the nature of the aforementioned database languages and the optimization thereof.

Using LEGOs to Teach Security Protocols

Lane Harrison, *University of North Carolina at Charlotte*

This project introduces exercises using a set of LEGOs and well-designed questions to increase the effectiveness of undergraduate information security courses. By using LEGO bricks, students are able to construct both classic and new protocols, as well as explore how the interactions between the two will affect the protocol's outcome.

Automatic Sign Generation in a Virtual World

Todd Heino, *University of the Pacific*

The ability to automatically generate signs within a virtual world is needed. This project finds a way to do this using distance table pathfinding and dynamic texture rendering. The end result is the ability to define a world using a text file and have signs appear in the 3D representation which guide a user when looking for certain points of interest.

A 16 Bit Microcoded Microprocessor for an FPGA

Steven Rubin, *Williams College*

A field programmable gate array (FPGA) is a chip that can be reconfigured with any circuit, thus making it an immensely powerful device for hardware developers and students. This project was the development of a microcoded microprocessor for an FPGA. The research included work in computer architecture and hardware description languages. A microcoded processor is especially useful to students because it allows them to design their own microcode for a given CPU. The existence of such a processor on an FPGA allows microcode development to proceed on physical hardware rather than an emulator.

Encouraging Design Reuse and Contribution in HCI Storyboarding

Lauren Cairco, *Winthrop University*

Storyboards for HCI product design are helpful, but traditional storyboarding is not a long-term solution because paper-based work is difficult to centralize and disseminate. To provide a more usable alternative, we developed a digital storyboarding environment that replicates traditional storyboard functionality but takes advantage of the sorting, storing, and organizing of information that technology enables. Designed for use with nontraditional displays to provide sufficient space and natural interaction, our application allows users to construct storyboards out of previously created features shown in the form of "cards" as well as to contribute their own cards to the library for future use.

Author Index

Abowd, Gregory D.	553	Bryfczynski, Samuel	168	Draper, Geoffrey M.	458
Adams, Elizabeth	399	Brynielsson, Joel	236	Eagle, Michael	321
Adams, Joel	143	Buck, Duane	256	Edwards, Stephen H.	173
Agrawal, Anukrati	291	Bulusu, Nirupama	34	Ellis, Heidi J. C.	134, 201
Aiken, Robert	197	Burg, Jennifer	473	Enbody, Richard J.	116
Albrecht, Jeannie R.	101	Cappos, Justin	111	Ericson, Barbara	86
Allevato, Anthony	173	Carroll, John M.	226	Ernst, Daniel	136
Almstrum, Vicki L.	201	Caspersen, Michael	487	Fairbrother, Dana	291
Altizer, Roger	534	Cassel, Lillian N.	64, 487	Falkner, Katrina	4
Anderson, Tom	111	Cavender, Anna C.	514	Fenwick Jr., James B.	296
Appel, Florence	333	Chan, Stephen C. F.	276, 504	Fisher, Kathleen	132
Armoni, Michal	443	Chang, George	91	Flatland, Robin Y.	387
Arshad, Naveed	372	Chao, Jie	158	Fletcher, George H. L.	260
Astrachan, Owen 192, 397, 483, 549		Cheek, Scott	296	Fox, Edward A.	69
Austing, Richard H.	64	Cheng, Yan	246	Frantz, C. Kolin.	163
Back, Godmar	453	Cheung, Joey C. Y.	504	Furst, Jacob David	468
Bailey, Mark	132	Cheung, Joey C. Y.	276	Gal-Ezer, Judith	126, 269
Baker, Alex	519	Cicirello, Vincent A.	286	Ganoe, Craig	226
Balch, Tucker	433	Clear, Tony	408	Gao, Yuanyuan	39
Barker, Lecia J.	153, 399	Clements, John	148	Garcia, Daniel D.	65
Barnes, Tiffany	199, 321	Clifton, Curtis	489	Gehringer, Edward F.	81
Barry, Frank E.	296	Cphoon, J. McGrath	158	Goldwasser, Michael H.	206
Bayliss, Jessica D.	199, 337	Collins, Carol B.	306	Greher, Gena R.	478
Beck, Barbara	539	Cooper, Stephen	67, 271	Gu, Song	39
Beck, Jon	201	Courte, Jill	29	Gundermann, Dawn	163
Bell, Tim	231	Craig, Michelle	221	Gutierrez, Ron	246
Bennett, Chris	76	Cromack, Jamie	199	Hadfield, Steven	251
Berque, Dave A.	485	Cuny, Jan	397	Haebe, Jessica	509
Beschastnikh, Ivan	111	Cupper, Robert D.	335	Hallstrom, Jason O.	524
Biggers, Maureen	9, 86, 163	Cutler, Robb	65	Hambrusch, Susanne ..	163, 183, 549
Binkley, David	163	Dale, Nell	197	Hamner, Emily	428
Bishop-Clark, Cathy	29	Dann, Wanda	271	Hanks, Brian	19, 24
Black, Michael D.	448	Davies, Gordon	487	Hansen, Stuart	377
Blaheta, Don	49	Davis, Janet	71, 551	Harper, Robert	132
Blank, Doug	433	de Lanerolle, Trishan	311	Harvey, Brian	136, 551
Boland, Michael G.	489	Deibel, Katherine	265, 333	Hasan, Sahar	178
Boustedt, Jonas	265	Denny, Paul	216	Haugan, Mark	183
Boyer, Kristy Elizabeth.	14	Deogun, Jitender S.	357	Hayes, Jenna	271
Brandt, Matt	24	Desai, Chetan	148	Hazzan, Orit	269
Brinkman, Bo	316	Dexter, Scott	134	Heeringa, Brent	483
Brown, Christopher	392	Diaz, Lien	397	Heines, Jesse M.	69, 478
Brown, Richard A.	106, 551	Dimond, Jill	86	Hewner, Mike	86
Brownfield, Andrew	413	DiSalvo, Betsy	86	Hirshfield, Stuart	335
Bruce, Kim	132	Docimo, Rebecca	301	Hislop, Gregory W.	134
Bruce, Rebecca F.	382	Dodds, Zachary	65	Hoffmann, Christoph	183
Bruckman, Amy	86	Donaldson, John L.	418	Horstmann, Cay S.	494

Horton, Diane	221	Lodder, Jerry	401	Nisan, Noam	443
Horwitz, Susan	163	Lopez, Mercedes	271	Nishida, Tomohiro	231
Hosking, Antony L.	183	Loveland, Kristin	178	Norris, Cindy	296, 413
Huggins, Jim	265	Lu, James J.	260	Nourbakhsh, Illah	428
Humphries, Thorna	423	Lurie, Andrei F.	494	O'Connor, Brian C.	438
Hundhausen, Christopher	291	Luxton-Reilly, Andrew	216	O'Hara, Keith	433
Huss-Lederman, Steven	163	Mache, Jens	34	Oldham, Joseph D.	333
Idosaka, Yukio	231	Mahar, Jan	226	Olsen, Gayle	539
Impagliazzo, John	197	Maheswaran, Muthucumaru	39	Omar, Kareem	403
Ioujanina, Anastassia	226	Malan, David J.	2	Paglione, Mike	301
Janzen, David S.	148	Maloney, John	2	Palmer, Edward	4
Jensen, Rachel	509	Malozemoff, Alexis	39	Panitz, Michael Panitz	204
Johnson, Menko B.	494	Maniyamaran, Balasubramaneyam	39	Paone, Timothy	226
Kacmarcik, Gary	281	Manley, Eric D.	357	Pargas, Roy P.	168
Kacmarcik, Sylvie Giral	281	Martin, C. Dianne	333	Parlante, Nick	483
Kahlon, Amardeep	399	Matthews, James R.	387	Pastel, Robert	392
Kaiser, Gail	178	McCartney, Robert	265	Peckham, Joan	67, 549
Kalahar, Kimberly	153	McCauley, Renée	19, 267	Pérez-Quñones, Manuel A.	173
Kalamkar, Vipul	306	McCullen, Mark	116	Pfaff, Ben	453
Kanemune, Susumu	231	McDowell, Charlie	153	Phillips, Robert	14
Kessler, Robert R.	458, 463, 534	McGettrick, Andrew	197, 267	Philpott, Anne	408
Khmelichek, Roman	246	McGill, Monica M.	347	Pivkina, Inna	401, 509
Kirst, Victoria	121	McKlin, Tom	86	Pontelli, Enrico	509
Koffman, Elliot B.	1, 64, 197	McMaster, Kirby	251	Poser, Susan	357
Korb, John T.	183	McWhorter, William I.	438	Presenter, The Mystery	265
Kourtchikov, Kirill	246	Meneely, Andrew	529	Preston, Jon A.	342
Krishnamurthy, Arvind	111	Meyer, George	59	Proulx, Viera K.	138
Kuhn, Sarah	478	Miller, Carolyn S.	81	Punch, William F.	116
Kumar, Amruth N.	499	Miller, Jessica	121	Purewal Jr., Tarsem S.	333
Kumar, Deepak	433	Moore, Brian	59	Pyster, Art	487
Kuno, Yasushi	231	Morelli, Ralph	311	Qin, Henry	271
Kurkovsky, Stan	44, 91	Morgan, Chris	403	Qin, Hong	188
Ladner, Richard E.	514	Moriyama, Etsuko	59	Ragonis, Noa	269
Lau, Winnie W. Y.	276, 504	Morreale, Patricia	91	Rague, Brian	251
Lauwers, Tom	428	Morrison, Briana B.	342	Raicu, Daniela Stan.	468
Lawrence, Andrea	399	Mundie, Craig	203	Ramsay, Stephen	59
LeBlanc, Richard	267	Munson, Ethan	163	Ranjan, Desh	401
Lee, Kyung Min.	403	Murphy, Christian	178	Raymond, Julie	39
Leisy, Jim	197	Murphy, Laurie	19	Rebelsky, Samuel A.	551
Leitner, Henry H.	2	Murphy, Tom	136	Reed, David	483
Leonard, Dana P.	524	Murtagh, Thomas P.	483	Reges, Stuart	132
Lester, James C.	14	Namiki, Mitaro	231	Reid, Karen	483
Letscher, David	206	Navarro, Emily	326	Reiser, Susan L.	382
Leutenegger, Scott Leutenegger	204	Nelson, Deborah	271	Richards, Brad	96
Lewis, Mark	204	Ng, Daniel	39	Riedesel, Charles P.	357
Lezin, Gaetjens	271	Ngai, Grace	276, 504	Riesenfeld, Richard F.	458
Liao, Sheng	39	Ni, Lijun	86, 433, 544	Roberts, Eric	65
Lin, Chi-Cheng	539	Nickens, Glenn V.	423	Rodger, Susan H.	163, 271, 403

Romano, Anthony	453	Stephenson, Chris	397	Vouk, Mladen A.....	14
Romney, Jason	473	Stepp, Marty	121	Walker, Daniel	433
Rosson, Mary Beth	226	Stone, Christopher A.	483	Walker, Henry	397
Roth, Robert I.	514	Stucki, David J.	256	Wallace, Scott A.	204
Rountree, Josh	296	Su, Jonathan	403	Wallis, Michael D.	14
Rusu, Adrian	301	Summet, Jay	433	Way, Thomas	362
Rusu, Amalia	301	Sung, Kelvin	204	Wein, Joel	246
Ryder, Barbara	163	Sweat, Monica	9	Weinberg, Jerry B.	54
Sahami, Mehran	483	Sweat, Monica	163	Westbrook, Suzanne	265
Salter, Richard M.	335, 418	Tabrizi, M. H. N.	306	Whalley, Jacqueline	408
Samal, Ashok	59	Talebi, Mujtaba	362	White, Curt M.	131
Sanderson, Donald B.	352	Taube-Schock, Craig	211	White, Laurie	331
Santiago, Confesor	301	Taylor, David Scot	494	White, William W.....	54
Schocken, Shimon	443	Taylor, Harriet G.	67, 69	Wick, Michael R.	367
Scott, Stephen	59	Thomas, William G.	59	Wiebe, Eric	403
Settle, Amber	549	Tira-Thompson, Ethan J.	423	Williams, Laurie	529
Shaikh, Reehan	39	Topi, Heikki	267, 487	Wittman, Barry	136
Sharma, Sean K.	494	Topol, Matthew	246	Wolz, Ursula	2, 199
Sharp, Alexa M.	335	Touretzky, David S.	423	Wrinn, Michael	136
Sheasley, Gretel	226	Towhidnejad, Massood	201	Wu, Zhen	158
Shell, Duane F.	59	Trevisan, Michael	291	Yang, Li	241
Sherman, Chris	246	Tucker, Allen B.	134	Yardi, Sarita	86
Simon, Beth	19, 265	Tucker, Ruth	271	Yilmaz, Tuba	9
Sinha, Hansa	226	Tyman, Damon	34	Yin, Edward C.	494
Sitaraman, Murali	524	Tymann, Paul T.	331	Young, Alison	65, 399
Slater, Don	271	Urban, Joseph E.....	69	Zander, Carol	19
Soh, Leen-Kiat	59	Urness, Timothy	76	Zhang, Mingrui	539
Spicer, Cole	296	van der Hoek, André	326, 519	Zur, Ela	126
Spradling, Carol L.	333	van Langeveld, Mark	463, 534		
Stephenson, Ben	211	Vilner, Tamar	126		