

# NSF CCLI & CPATH Project Showcase

SIGCSE 2010  
March 10 - 13  
Milwaukee, Wisconsin



**National Science Foundation**  
CCLI & CPATH Showcase



## Program at a Glance

### Thursday, 10:00 a.m.—11:30 a.m.

- PREOP (Providing Robotic Experiences Through OO Programming)** 4  
Monica Anderson, et al., University of Alabama Tuscaloosa
- Using Proplets to Learn Introductory Programming Concepts by Solving Problems** 5  
Amruth Kumar, Ramapo College of New Jersey
- Immersive Security Education Environment Using Second Life** 6  
Angsana Techatassanasoontorn, Pennsylvania State University Park
- MOBILE—A Portable Networked Laptop Instructional Environment** 7  
Paul Wagner, University of Wisconsin-Eau Claire

### Thursday, 3:00 p.m.—4:30 p.m.

- A Virtual Reality Laboratory and Curriculum for Undergraduates** 8  
Daniel Cliburn, Michael Doherty, et. al., University of the Pacific
- UPass—Usable Password Authentication Simulator for Students** 9  
Binto George & Hema Sandeep Narava, Western Illinois University
- APOGEE: An Instant Grading Tool for Sophisticated Course Projects** 10  
Xiang Fu, Kai Qian, & Boris Peltsverger, Hofstra University
- Computational Thinking across the Curriculum (CPATH)** 11  
Amber Settle & Ljubomir Perkovic, DePaul University

### Friday 10:00 a.m.—11:30 a.m.

- Modern Distributed Computing Education** 12  
Barry Kurtz & Rahman Tashakkori, Appalachian State University
- Promoting Algorithm Development in Introductory CS Classes** 13  
Jungsoon Yoo, Sung Yoo, et. al., Middle Tennessee State University
- MediaScripting: Media Computation with Functional Programming, Scripting, and Design Principles** 14  
Samuel Rebelsky, Grinnell College
- Applied CS for the Humanities and Social Sciences (CPATH)** 15  
Steve Reiss, Tom Doepfner, et. al., Brown University

## Program at a Glance

### Friday 2:45 p.m.—4:15 p.m.

- SoftHum: Student Participation in the Community of Open Source Software for Humanity** 16  
Heidi Ellis, Gregory Hislop, & Ralph Moprelli, Trinity College
- Increasing Attractiveness of Computing: The Design and Evolution of Introductory Computing Coursework that Elicits Creativity** 17  
Eric Freudenthal & John Fernandez, University of Texas at El Paso
- Active Learning for the Transformation of the Undergraduate Experience (CPATH)** 18  
Cindy Grimm, Washington University (Missouri)
- Animated Database Courseware (ADbC)** 19  
Mario Guimaraes & Meg Murray, Kennesaw State University

### Saturday 10:10 a.m.—11:40 a.m.

- Computational Thinking Showcase: Computing Concepts Across the Curriculum (CPATH)** 20  
Vicki Allan, Donald Cooley, Chad Mano & Russ Weeks, Utah State
- Information Security: HIPAA Case Study and Security Workbook** 21  
Susan Lincke, University of Wisconsin-Parkside
- Indexed Captioned Searchable Videos** 22  
Jaspal Subhlok, Richard Alo, et. al., University of Houston, Downtown
- Building Security In: Injecting Security throughout the Undergraduate Computing Curriculum** 23  
Blair Taylor & Siddharth Kaza, Towson University

## PREOP (Providing Robotic Experiences Through Object-Based Programming)

Thursday, 10:00 a.m.—11:30 a.m.

**Monica Anderson, Susan Vrbsky, Marcus Brown, Briana Lowe Wellman, Andrew McKenzie, James Davis & Karl Smith;** University of Alabama Tuscaloosa

---

The demand for computer scientists is expected to continue to increase irrespective of the current state of the economy. Unfortunately, the supply is not expected to match the demand as the number of computer science majors has decreased substantially since the year 2000. As a result, universities and colleges are attempting to identify new ways to attract and retain prospective students into the field of computer science in order to increase the number of majors. PREOP (Providing Robotic Experiences through Object-Based Programming) is an approach that combines the Alice interface and robots for a CS1 Laboratory. PREOP is an interactive 3D animation programming environment, that allows students to program real robots using a drag-and-drop, syntax-free interface. The goal is to foster student motivation and increase student understanding of the fundamental concepts within the first-year curriculum. Initial results indicate that the students in the PREOP Lab who are eligible for CS2 are more likely to rate their skills and knowledge above average than the students in the non-PREOP Labs, and more likely to be registered for the CS2 course than the students in the non-PREOP Labs.



## Using Proplets to Learn Introductory Programming Concepts by

Thursday, 10:00 a.m.—11:30 a.m.

Amruth Kumar; Ramapo College of New Jersey

---

Proplets are web-based tutors designed to help *Computer Science I* students learn programming concepts by solving problems. While working with proplets, students repeatedly solve problems, and get instant feedback on their solution as well as explanation of the correct solution. Each proplet assesses the student's prior knowledge and presents problems on only those concepts that the student does not already know. This feature minimizes the number of problems solved by the student while maximizing the student's learning. The feedback provided by proplets after each problem has been proven in controlled evaluations to help students learn.

Each semester, instructors request proplets for their class by sending an email – no class list is needed. In response, a dedicated web site is set up for each instructor. The instructor then simply directs students to use proplets at this site, either as closed lab exercises, or as after-class assignments. After students have used each proplet, instructors can obtain a report as an Excel spreadsheet. Currently, proplets are available for arithmetic, relational, logical and assignment expressions, if-else and switch statements, while and for loops, functions and C++ pointers. They are available for C, C++, Java, and C#. Proplets have been adopted by dozens of faculty, and are currently being used by hundreds of students each semester. They are free for educational use. They run in any Java-enabled browser – no software installation is necessary. For additional information, and to try out proplets, please visit <http://www.proplets.org>. If interested in adopting proplets, please contact Amruth Kumar, [amruth@ramapo.edu](mailto:amruth@ramapo.edu)

**Acknowledgments:** This work was supported in part by the National Science Foundation under grant DUE-0817187.

## Immersive Security Education Environment Using Second Life

Thursday, 10:00 a.m.—11:30 a.m.

Angsana Techatassanasoontorn; Penn State University Park

Security education has traditionally focused on the technical aspect of security by using abstract concepts that are not situated in real-life contexts. Using an integrated Web and Second Life environment, this project develops a scenario-based security learning module to help individuals obtain a basic understanding and appropriate skills to protect themselves against key security threats. The set up involves several teams competing against each other to sell products online. In particular, each team is tasked to open a virtual storefront in Second Life using a limited amount of money. They are also required to use an appropriate router configuration and other security mechanisms to protect their information assets. Several game-like activities are embedded in the learning module. For example, teams are required to do a scavenger hunt and take quizzes with passing grades to earn the right to use security mechanisms on their web sites. Various attacks are also launched against teams that have weak security setup (e.g., open ports, weak password, and a lack of appropriate access control). The assessment results using various pre- and post-measures show improvement in security learning. Visit <http://isee.ist.psu.edu> for more information about the project.



## **MOBILE—A Portable Networked Laptop Instructional Environment**

**Thursday, 10:00 a.m.—11:30 a.m.**

**Paul Wagner; University of Wisconsin – Eau Claire**

---

The MOBILE project at the University of Wisconsin – Eau Claire aims to develop a mobile/portable software system for delivering hands-on computer intensive educational workshops in any area of computer science and beyond. The project is funded by a National Science Foundation (NSF) Course, Curriculum and Laboratory Improvement (CCLI) Phase 2 grant (NSF #0817295) awarded in 2008. MOBILE will eventually be distributed as an open-source software system.

The three major goals of the MOBILE project are to:

1. Develop a mobile wireless laptop-based computer network management system that can be used in a variety of computer science (and other) educational workshop scenarios and can be quickly installed on participants' laptop computers from a portable storage device (such as a USB drive) given a minimal subset of memory, disk and software requirements,
2. Package this functionality with an easy-to-use interface that can be used by content developers and workshop presenters, and
3. Demonstrate the value of this system to educators by presenting mobile hands-on computer science workshops to educators as a demonstration of the power of this system and holding workshops teaching educators how to use this system for their own computer science education workshops.

At SIGCSE 2010, we will demonstrate how to create a workshop scenario (in either computer security or a CS1-style programming laboratory) using MOBILE, how to start the management/instructor side of the system, and how workshop participants can connect to a MOBILE session and interact with both the instructor and other participants. We look forward to suggestions from showcase visitors as to features which would make the MOBILE system useful to them for their educational work.



## A Virtual Reality Laboratory and Curriculum for Undergraduates

**Thursday, 3:00 p.m.—4:30 p.m.**

**Daniel Cliburn, Michael Doherty, James Miller & Stacy Rilea;**  
University of the Pacific

---

Virtual reality (VR) is a contemporary area of study with numerous applications in art, training, simulation, and health care. Unfortunately, most undergraduate Computer



Science (CS) students receive little exposure to this exciting field. The omission of VR from the undergraduate CS curriculum may be attributed to reasons such as a lack of easily accessible instructional materials, an already overcrowded curriculum, and the previously high cost of appropriate equipment. In this project, we have begun a number of initiatives to address these issues. Specifically, we have developed an undergraduate VR laboratory with economically priced devices capable of generating stereo images, producing force feedback, and tracking user movement. In addition, we have developed drafts of four flexible online lesson units chosen from the Computing Curricula 2001 recommendations for an elective course on virtual reality: "User interface issues in virtual reality", "Stereo projections", "Force feedback simulation, haptic devices", and "Applications of virtual reality." The lesson units have been incorporated into several undergraduate courses, including Computer Graphics and Human-Computer Interface Design (for computer science majors) and a general education course for non-majors. Our evaluation of these units suggests that they do allow us to achieve the goal of teaching virtual reality concepts to undergraduate students without disrupting curricular structures.



## **UPass—Usable Password Authentication Simulator for Students**

**Thursday, 3:00 p.m.—4:30 p.m.**

**Binto George & Hema Sandeep Narava; Western Illinois Univ.**

---

Computer Security courses need to emphasize usable security concepts in order to prepare students for designing secure systems with adequate safeguards against vulnerabilities arising from the lack of usability and the exploitation of human factors. Traditionally, computer security professionals focus on technical aspects of security, leaving user interface design to GUI professionals. However, usable security requires a different design standard that GUI professionals rarely follow. Thus, security professionals, being the stakeholders, should develop a deeper understanding of the design process for ensuring usable security. From our experience, in a typical undergraduate computer security course, students may not find enough time to design and develop a full-fledged system from scratch while performing threat analysis and usability testing.

In our view, password authentication is a usable security relevant function with many real life applications. Therefore, we develop Usable Password Authentication Simulator for Students (UPass) that helps students visualize the user authentication process from three different perspectives, namely, Defense, Use and Offense. UPass, currently under development, provides a number of built-in authentication schemes such as text passwords, cognitive passwords and graphical passwords. Once fully developed, UPass will provide a platform for students to experiment with different authentication mechanisms, tryout password policies and study the security-usability impact. UPass has fully open source implementation using Java and Google Web Development Toolkit (GWT) that allows for the addition of new authentication schemes in the future. This work is supported, in part, by the National Science Foundation CCLI Grant (DUE-0736643). Project Web Page: <http://www.wiu.edu/UsableSecurity>.

## **APOGEE: An Instant Grading Tool for Sophisticated Course Projects**

**Thursday, 3:00 p.m.—4:30 p.m.**

**Xiang Fu, Kai Qian, & Boris Peltsverger;** Hofstra University, Southern Polytechnic State University & Georgia Southwestern State University

---

Sophisticated course projects are frequently used to expose students to the complexity of software development. This raises new challenges for automated grading systems to handle graphical user interface, backend database and asynchronous data exchange.

Supported by NSF DUE awards 0836859, 0837275 and 0837020, APOGEE (Automated Project Grading and Rapid Feedback System) is an automated grader for realizing a trial-and-failure learning strategy. The idea is simple: let students try project submission as many times as they want before the project deadline. For each submission, a thorough inspection will be performed to make sure that it accomplishes not only the functional requirements but also many desired quality attributes such as robustness and security. APOGEE provides a powerful feature set for instructors. Project testing scripts can be easily generated by composing testing actions in a web based GUI editor. It also provides two project reporting services to students, e.g., step-by-step animation re-play of failed test cases.

A collection of project-oriented course modules have been developed for APOGEE. They are called APOGEE Packs, because they can be easily loaded into any APOGEE server. Each APOGEE Pack is self-contained and consists of a project specification, a partial implementation for students and a collection of predefined test cases. APOGEE and its course packs are freely available at [http://people.hofstra.edu/Xiang\\_Fu/XiangFu/Projects/APOGEEHomepage/index.php](http://people.hofstra.edu/Xiang_Fu/XiangFu/Projects/APOGEEHomepage/index.php).

## Computational Thinking across the Curriculum (CPATH)

**Thursday, 3:00 p.m.—4:30 p.m.**

**Amber Settle & Ljubomir Perkovic; DePaul University**

---

In this project we are developing a framework for implementing computational thinking as a fundamental skill in the context of undergraduate education. Specifically we focus on using liberal studies courses -- part of the education of the vast majority of undergraduates -- as a vehicle for the teaching of computational thinking. A total of 18 DePaul University faculty, 7 from the School of Computing, 3 from the School of Cinema and Interactive Media, and 8 from the College of Liberal Arts and Sciences, are collaborating on the project. We are working toward achieving a consensus on the role of computational thinking in the liberal studies curriculum and on the best methods to incorporate it. The framework we are developing is designed for faculty without formal training in information technology, for use in understanding and integrating computational thinking into their liberal studies courses. It includes examples of computational thinking in 19 general education courses taught in 5 different domains in the Liberal Studies Program, as well as sample in-class activities, assignments, and other assessments for the courses. The examples in the different courses are tied together using terminology defined by Peter Denning in his Great Principles of Computing project, so that similar types of computational thinking can be seen in a variety of specific contexts. This categorization also aids the understanding of the computational thinking found in the courses and provides a template for future work. More information about the framework can be found at the project web site: <http://compthink.cs.depaul.edu/>

## Modern Distributed Computing Education

**Friday, 10:00 a.m.—11:30 a.m.**

**Barry Kurtz & Rahman Tashakkori;** Appalachian State University

---

We are developing educational materials to support instruction in both distributed memory computing, such as using MPI or web services, and shared memory computing, such as using OpenMP or threads. Rather than teach these topics in isolation we are using a “modern” approach where we integrate all forms of distributed/parallel computing in the solution of computing problems. We organized our activities into two phases: (1) during year one material was developed as part of a complete course that was taught in the summer 2008 and spring 2009 and (2) during year two materials are being modularized and introduced into existing courses as separate learning modules. We use an approach to active learning where Microlabs provide hands-on experience by asking students to complete short lab activities (typically 5 to 15 minutes each) during lecture.

Our materials cover multi-core computing using threads in Java or C#, using OpenMP in C or the Task Parallel Library in C#, using MPI either in the Linux environment or the .NET environment, and deploying web services using either Java or C#. We have developed a repository of materials developed directly as part of this grant and including materials from courses taught by other researchers, such as the Parallel Computing and Grid Computing Courses taught by Dr. Barry Wilkinson at UNC Charlotte and the Distributed Systems course taught by Yaohang Li at North Carolina A&T. Several regional workshops were conducted as part of this project.

The repository can be accessed at  
<http://www.cs.appstate.edu/distributedComputing>

## Promoting Algorithm Development in Introductory CS Classes

**Friday, 10:00 a.m.—11:30 a.m.**

**J. Yoo, S. Yoo, Z. Dong, C. Pettey, S. Seo & Z. Dong;** Middle Tennessee State University

---

Beginning students in the Computer Science curriculum frequently do not appreciate the value of the design phase of any programming project. In addition, when asked to develop an algorithm for a given problem, they have difficulty distinguishing between a general solution and a completed program. As with other skills, what is needed is a way to practice applying the basic concepts of algorithm design. To that end, we have developed AlgoTutor (The Algorithm Tutor), an online, graphical tool for practicing algorithm development. AlgoTutor is an interactive tool that provides automatic assessment of student designed algorithms. It supports both top-down and bottom-up design methodologies. In addition to the student algorithm design interface, there is an instructor interface that allows the teacher to create/edit problems and their solutions, manage student accounts, and analyze student grades as well as student activities.

## **MediaScripting: Media Computation with Func. Prog., Scripting, and Design Principles**

**Friday, 10:00 a.m.—11:30 a.m.**

**Samuel Rebelsky; Grinnell College**

---

Media computation in CS1 has proven successful in attracting and retaining students in a variety of contexts. In the MediaScripting project students write Scheme scripts to create and manipulate images within the GIMP (GNU Image Manipulation Program), an open-source raster graphics application.

An important aspect of the MediaScripting class is a project in which students write algorithms to generate a collection of related images, based only on the size and an integer. The resulting works are judged both by computer science faculty (for the design of the algorithm) and by art faculty (for the design of the image).

Because they script within the GIMP, students are able to use a mixed-mode approach to image creation and manipulation: the both write scripts and do things "by hand". This mixed-mode approach helps motivate programming problems; students find that they want to write scripts to replace what they do by hand - to ensure precision, to encode a sequence of actions, and even to experiment. Students discover that they can do things with scripting that would not be otherwise possible such as building brushes that change their shape or color over time, and more. Scripting an application presents opportunities not available with a typical set of drawing primitives.

Scheme allows us to have students think about images in different ways. They consider everything from an imperative approach to scripting the tools, through a pure and hierarchical shape-based model, to higher order approaches, such as modeling images as functions from positions to colors.

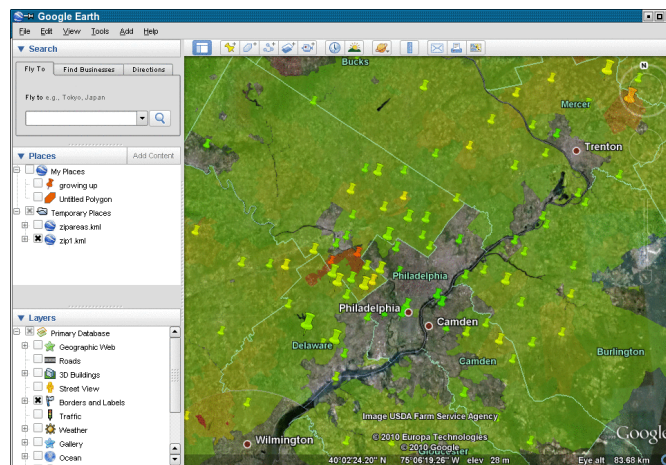
## Applied CS for the Humanities and Social Sciences (CPATH)

Friday, 10:00 a.m.—11:30 a.m.

Steve Reiss, Tom Doeppner, John Hughes & Shriram Krishnamurthi; Brown University

---

Computation and computational thinking are increasingly needed in all fields, not just the sciences. Our project involves developing an integrated and rigorous sequence of courses that teach students in the humanities and social sciences the computational skills they will need to prosper in future jobs or research. Based on exploratory work involving mini-courses and a small set of students, we taught the first course in our proposed sequence last fall. This course had the students address three specific problems, the first involving politics, the second textual analysis, and the third sociology. The course covered topics such as data gathering, data validation, data analysis, statistics, and data presentation and taught skills such as the use of Excel, creating data-oriented web pages, python programming, and geographical information systems. Students identified and then addressed their own problems using the skills they learned in group final projects. Information about the course is available at <http://www.cs.brown.edu/courses/csci0931>.





## **SoftHum: Student Participation in the Community of OSS for Humanity**

**Friday, 2:45 p.m.—4:15 p.m.**

**Heidi Ellis, Greg Hislop & Ralph Morelli; Trinity College**

---

SoftHum champions the use of Humanitarian Free and Open Source Software (HFOSS) as a foundation for computing education that can attract students to the major, provide an excellent learning vehicle, and achieve social benefits. SoftHum builds on the foundation of the Humanitarian FOSS project ([hfoss.org](http://hfoss.org)), funded by the NSF CISE CPATH program to revitalize computing by engaging undergraduates in building open source software for humanitarian organizations.

The SoftHum project has three main goals:

- To develop a process to support student success in a community-based software experience using HFOSS,
- To document an environment that supports student open source experience, and
- To disseminate results of projects and position additional faculty to apply HFOSS in computing education.

SoftHum activities have included surveying both students and instructors involved in HFOSS projects in the classroom, the construction of a web site that provides support and guidance to instructors and others who are teaching or supporting students involved in HFOSS projects, and a workshop for educators focusing on preparing faculty members to teach courses in which students participate in HFOSS software development projects.

Initial results from the SoftHum project indicate that the humanitarian aspect of computing shows signs of attracting students to the major. Another significant result is the identification of several software processes that can be applied within a course based on HFOSS. Future plans include the continuation of surveying of students and instructors, and a follow-on summer workshop to further and refine the course materials to support HFOSS in the classroom.

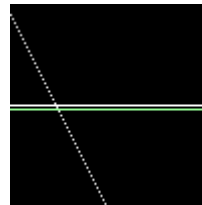
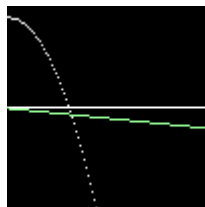
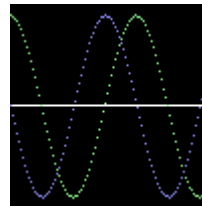
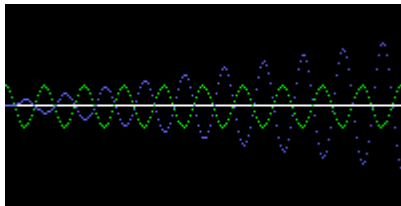
## Increasing Attractiveness of Computing

Friday, 2:45 p.m.—4:15 p.m.

**Eric Freudenthal & John Fernandez;** University of Texas, El Paso

---

This CCLI-funded effort initially focused upon the integration of programming into a required entering students program in order to provide early experiences relevant to choosing CS as an academic major. Early in this project, we observed that graphically oriented introductory programming assignments could be structured in a manner that also strengthened students' understandings of algebra, trigonometry, ballistics, and resonance. The course now introduces a range of entering students to "programmed computation" in a manner that prepares them to select among multiple STEM majors. Variants are being taught to entering students from across engineering, integrated into introductory programming courses offered to students from multiple STEM majors, and pilot experiments are being conducted within high schools.



## **Active Learning for the Transformation of the Undergraduate Experience (CPATH)**

**Friday, 2:45 p.m.—4:15 p.m.**

**Cindy Grimm; Washington University**

---

Washington University has focused on transforming the entire computer science undergraduate curriculum to an active learning approach. We classified courses into two categories, Studio (more project and programming-based) and Fundamental (theory). The Studio classes have adopted a critique-based, team-focused approach with a minimum of lecture. The Fundamental classes have reduced in-class lecture time and replaced it with in-class exercises. We have nearly completed the transformation, with our first students to pass through the modified curriculum moving into their junior year.

## **Animated Database Courseware (ADbC)**

**Friday, 2:45 p.m.—4:15 p.m.**

**Mario Guimaraes & Meg Murray; Kennesaw University**

---

The Animated Database Courseware (ADbC) project consists of 4 modules and 20 sub-modules including more than 100 programs (tutorials, exercises and animations). Current modules available cover the areas of database design, interactive SQL, stored procedures and triggers, transactions and database security. The software has been made freely available and is located on the Web at <http://adbc.kennesaw.edu>. The current version of ADbC as well as the recent updates as a result of evaluation and usability tests will be displayed. Recent upgrades include a re-design of the interface, refining and increasing the number of animations of each sub-module and prototypes of backup / restore, startup / shutdown and datawarehouse.

## Computational Thinking Showcase: Computing Concepts Across the Curriculum

**Saturday, 10:10 a.m.—11:40 p.m.**

**Vicki Allan, Donald Cooley, Chad Mano & Russ Weeks;** Utah State University

---

The goal of our CPATH grant is to create a set of Interactive Learning Modules (ILMs) which capture interesting computer science problems and challenge users to think creatively. Rather than employing a syntax first approach, we create stand-alone modules which teach computer science using experiential learning. Many students know little about computer science, and even after taking an introductory course, view computer science as “just programming”. Our goal is to introduce intriguing problems and basic underlying concepts which computer scientists address, and allow students to experiment with tools built on such computational thinking in an interactive and fun environment.

We have developed numerous ILMs for use by students and teachers at the K-12 and university level. ILMs range from teaching basic computer science principals such as estimation, looping, and abstraction to demonstrating graph traversal and hashing methods. The ILMs have been employed as outreach in the middle school, as an introduction to computer science in general education classes, and within our undergraduate curriculum in computer science.

Due to an RET supplement, four area teachers have spent time designing ILMs for their classrooms. The csilm system allows teachers to copy lessons into their private classroom space and easily modify the instructions and presentation order. It is flexible enough to allow us to build lessons out of existing internet resources as well as ILMs produced by a team of graduate and undergraduate students, some completing ILM projects as part of a capstone class.

Join us for a demonstration of existing resources and a discussion of future possibilities.

## Information Security: HIPAA Case Study and Security Workbook

**Saturday, 10:10 a.m.—11:40 p.m.**

**Susan Lincke;** University of Wisconsin-Parkside

---

Information systems security is complex because attackers need only find one vulnerability, while defenders need to close all holes. This course combines a business-oriented, risk-based approach with technology to plan security from the high-level down. This approach focuses on where risk lies and considers all options: technical, administrative, and physical. Career opportunities in IS security include security analyst and IS auditor, for those with technical and business expertise.

Working with authentic problems enables students to practice the skills they need in the workplace. Service learning enables students (and faculty) to gain valuable experience while contributing to their community. This Information Systems Security course enables both benefits, by providing a case study as well as a Small Business Security Workbook. Students practice first with the case study, and then apply skills with community partners. (It is possible to use the case study without the service learning aspect.)

The case study involves a small doctors' office, which needs to adhere to the privacy and security requirements of HIPAA (Health Insurance Portability and Accountability Act). The security workbook leads students through a design of a security system for a small to medium-size business. Topics in the workbook include policy development, risk and business impact analysis, information security, network security, physical security, and incident response. Materials for the course are taken from best practices, including COBIT, Certified Information Systems Auditor (CISA), Certified Information Security Manager (CISM), and Certified Information Systems Security Professional (CISSP). These sources enable students to be one step closer to professional certification.

## **Indexed Captioned Searchable Videos**

**Saturday, 10:10 a.m.—11:40 p.m.**

**Jaspal Subhlok, Richard Alo & Lecia Barker;** University of Houston - Downtown

---

The vision behind ICS videos project is flexible STEM coursework. The project is based on Tablet PC videos, which allow free mixing of prepared (PowerPoint) viewgraphs and hand annotations. Prior research shows that video lectures are a powerful and versatile resource, but suffer from the inability to quickly access the content of interest. The goal of this project is to improve access to information in videos by adding indexing, captioning, and search capability. The ICS videos framework consists of 1) Index points: The recorded video lectures will have a list of logical index points, each in the form of a snapshot representing a video segment; 2) Text captions: The entire audio stream, including the lecture and student interaction, will be presented as text in a separate panel; and 3) Keyword search: Video segments that contain search keywords in the audio will be displayed, along with information on keyword frequency. The key technical challenge in the project is automatic generation of ICS videos. A rigorous evaluation is being conducted in a range of disciplines with 1000s of students. Deployment of ICS videos spans Computer Science, Biology & Biochemistry, and Geosciences at University of Houston Central campus and UH Downtown. Evaluation of captioning for deaf and hearing-impaired students will be conducted at the Texas School for the Deaf (TSD). The framework is made freely available to educational institutions.



## Building Security In: Injecting Security throughout the Undergraduate Curriculum

**Saturday, 10:10 a.m.—11:40 p.m.**

**Blair Taylor & Siddharth Kaza; Towson University**

---

Despite its clear and growing importance, security education is often relegated to a secondary role in undergraduate curricula. Exposure to computer security concerns is limited to specialized courses and tracks that reach a small percentage of students, late in their academic careers. A successful approach must engage more students earlier in their education and adapt to fit the needs of different educational institutions. Our project (<http://triton.towson.edu/~cssecinj>) integrates security principles beginning with the core courses (CS0, CS1, and CS2) and reiterates these concepts in upper-level undergraduate courses. The primary components of our project are:

- **Security Injections Modules:** Each module is modeled after the traditional scientific lab and includes a background section, laboratory exercise, discussion questions, and a security checklist. Modules for the core classes are focused on the "big three" security vulnerabilities: integer overflow/errors, buffer overflow, and input validation.
- **Collaborative development and deployment:** Each module has been developed through an intense collaborative process with our partners at institutions ranging from community colleges to comprehensive universities using a Develop-Train-Pilot-Revise-Deploy model.

We have obtained promising results with the development of twelve modules deployed across five institutions. More than 1000 students have been exposed to our modules through lab exercises and extra credit assignments. Exit surveys on graduating seniors show significantly higher security awareness scores for students who have used the injection modules. Further evaluation and module development are in progress. By next year, the modules will be deployed and evaluated in classes at over 15 institutions in Maryland.



The NSF Showcase is an annual event at the SIGCSE conference. If you are working on an NSF CCLI or CPATH grant, or have recently completed one and would like to present at the showcase, please contact Aaron Bloomfield at [aaron@virginia.edu](mailto:aaron@virginia.edu). The selection process will begin in the fall.

<http://www.cs.virginia.edu/~asb/nsfclishowcase>